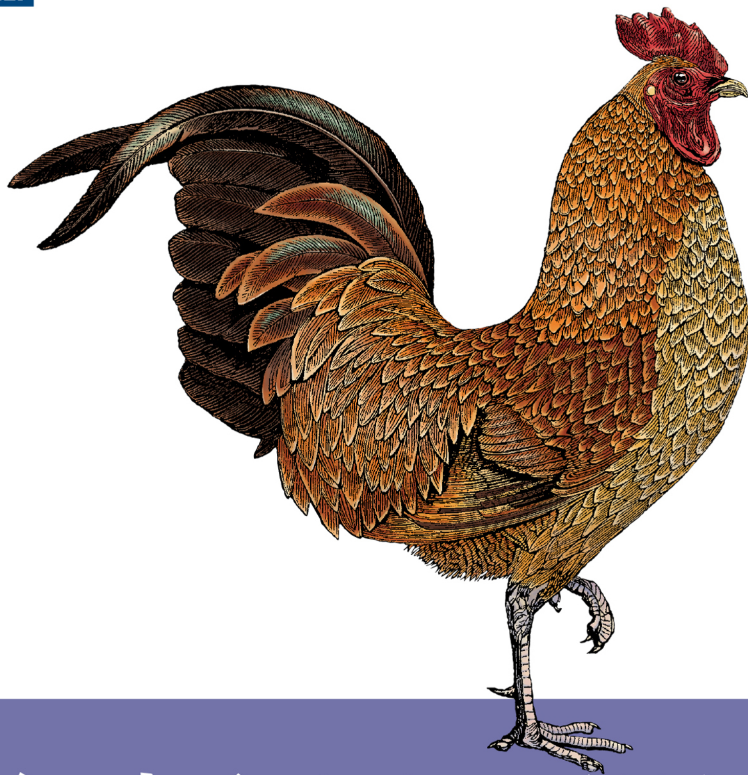
O'REILLY®



图灵交互设计丛书

第2版



# 移动应用UI设计模式

Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps

[美] Theresa Neil 著
[美] Jenifer Tidwell 序
田原 译

人民邮电出版社
POSTS & TELECOM PRESS

# 数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

## 译者介绍



### 田原

毕业于北京理工大学电子工程系，现在互联网公司任职产品经理，主要从事基于社交平台的企业级应用和移动应用设计。兴趣广泛，时而臭美，时而极客。擅长翻译科技类著作。




图灵交互设计丛书

# 移动应用UI设计模式（第2版）

## Mobile Design Pattern Gallery
Second Edition
## UI Patterns for Smartphone Apps

[美] Theresa Neil 著
田原 译

O'REILLY®

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

O'Reilly Media, Inc.授权人民邮电出版社出版

人 民 邮 电 出 版 社
北　　京

图书在版编目（CIP）数据

移动应用UI设计模式 / （美）尼尔（Neil,T.）著 ；
田原译. -- 2版. -- 北京 : 人民邮电出版社, 2015.1
（图灵交互设计丛书）
ISBN 978-7-115-37790-6

Ⅰ. ①移… Ⅱ. ①尼… ②田… Ⅲ. ①移动终端—应
用程序—程序设计 Ⅳ. ①TN929.53

中国版本图书馆CIP数据核字(2014)第282340号

## 内 容 提 要

这是一本移动应用 UI 设计模式参考书，分 11 大类介绍了时下安卓、iOS 以及 Windows Phone 平台上 90 多种移动应用设计模式（包括反模式），用 1000 张屏幕截图和图解帮助读者理解和利用这些模式，以解决常见的设计难题，同时提供了即学即用式的技巧和经验。

本书适合创造移动应用的产品经理、开发人员和 UI 设计师阅读。

◆ 著　[美] Theresa Neil
译　田　原
责任编辑　岳新欣
责任印制　杨林杰

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京　印刷

◆ 开本：880×1230　1/24
印张：13.75
字数：278千字　2015年1月第2版
印数：1－5 000册　2015年1月北京第1次印刷

著作权合同登记号 图字：01-2014-7516号

定价：79.00元

**读者服务热线：(010)51095186转600　印装质量热线：(010)81055316**

**反盗版热线：(010)81055315**

**广告经营许可证：京崇工商广字第 0021 号**

# 版权声明

© 2014 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2015. Authorized translation of the English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2014。

简体中文版由人民邮电出版社出版，2015。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

# O’Reilly Media, Inc.介绍

O’Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O’Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O’Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O’Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O’Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O’Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O’Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O’Reilly Radar 博客有口皆碑。”

——*Wired*

“O’Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O’Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O’Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

# 目录

# 序

说出一样事物的名称，往往是了解它的开始。

我 5 岁大的儿子和许多孩子一样，喜欢看天上的云。最近，他慢慢认识到不同种类的云有着不同的名称。天分使然，他开始分门别类地记忆这些云——卷云、积云、层云、卷层云、积雨云、高层云、荚状云……这其中包括我知道的所有名称，还有一些我不知道的。“浓积云”这个名称我之前肯定闻所未闻。

现在，他看着天空，能把云的分类说得明明白白。不仅如此，他比以前更加注意其中细微的差别。他学会了通过观察云的质感、颜色、高度、运动等，辨别云的类型（当然，不是总能轻易做到这一点，但也没有被难住）。他可以预测哪片云带雨，哪片云不带雨，有时候还挺准的。

他以学龄前儿童的方式，运用关于云的知识分析出大致的天气变化。他会说：“卷积云可能意味着天气转暖。”或者说：“浓积云也许会变成积雨云！那样的话，我们就会遭遇暴风。”

最重要的是，知道这些云的名称使他非常快乐。似乎所有的小孩子都会为能叫出他们喜欢之物的名字而兴奋不已，无论那是云、恐龙、昆虫、汽车、洋娃娃，还是电影人物。当然，他们的想象力并不受左脑思维方式所限，尽管我们成年人认为这不切实际——虽然我儿子称那些云为“积云”，但这并不妨碍他在云里看到宫殿、鸭子和花椰菜。

说说我们成年人吧。这就涉及本书的主题：通过识别界面设计模式并为其命名，我们可以更好地“看”这些界面。由于我们的大脑更加清楚我们想要寻找什么，我们会注意到更多的细节。当我们知道某个界面模式会如何表现时，就能预测所用的软件将如何运作。然后，我们可以用一个贴切的新词向别人描述我们看到的事物。

那么怎么学会这些设计模式呢？

我儿子在认识云时用到的最好的工具是图片，大量的图片。他在书里和网站上观察这些“样本”之后，学会了分辨不同类型的云之间极为细微的差异，其中一些细微到很难用言语描述。

同样，学习界面设计模式的最佳方法是观看图例。我是个作家，当然热爱文字。如果不拘礼节，我能滔滔不绝地讲解设计模式的定义，选择模式的原则，以及它们之间的差别。但我知道，任何一个只想着设计界面的人，也就是任何一个想把设计模式当作一部分行业知识的人，真的不需要了解所有这些内容。对于一种模式，他们只需要恰到好处的说明来领会它，之后需要看到一系列精挑细选的生动示例来巩固和吸收这些知识。

在这本书里，Theresa Neil 汇集了大量的设计模式图片。根据我自身的经验，我可以说这项工作之艰巨难以想象。评测这么多的移动应用、找出其中优秀的设计，再把所有这些经过仔细编目的屏幕截图进行分类，真可谓不小的壮举。

对移动界面设计师来说，这本书弥足珍贵。愿意的话，你可以从头到尾读一遍，但更重要的是，你可以利用其中的示例改进自己的设计。

判断这些示例中哪些设计是合理的，然后想一想哪些最适合你自己正在设计的应用。

把这本书当作设计灵感的源泉。在与设计模式本身无关的方面，比如在图标设计和色彩运用上，我也很欣赏这些设计范例。

用这本书来扩展你的知识，了解现有应用的工作原理，而不用费时费力地下载并使用所有这些应用（别忘了，是在多种设备上使用）。

看完这本书，你甚至可能到日常使用的移动应用中寻找你自己的设计模式示例。实际上，我敢打赌，一旦你熟悉了这些模式的名称，就一定会这样做。在让我儿子在野外指出几次“浓积云”之后，我对它已经非常熟悉了。天哪，真不知道以前没有了解这个知识的时候我是怎么活着的。

好好享受阅读这本书的乐趣吧！

——Jenifer Tidwell

# 前言

有时候，停下来思考一下影响可用性设计的诸多因素是件好事。但是往往时间不允许，你只能挽起袖子着手工作。本书正适用于这种情况。

一方面，自本书第 1 版于 2011 年出版以来，移动领域已经发生了巨大的变化。在第 1 版中，我写到了六个移动操作系统，而其中三个——WebOS、塞班（Symbian）和黑莓（BlackBerry），在当下的移动领域里已经失去了竞争力。

另一方面，变化又没有那么大：本书第 1 版里写到的 70 多个设计模式，大多数仍在使用，只增加了屈指可数的几个新模式。然而，最新的设计模式更多地展现了“以移动为中心”的理念。设计师们终于不再埋首桌面端和网页端隐喻，而从根本上寻找移动界面的设计方案。我希望这种状况持续下去，并能加快前进的步伐。

另一个变化：2011 年，我对与操作系统无关的设计也持乐观态度，这意味着我们作为设计师和开发者可以创造适用于多平台的界面。实际上，事实正好相反，iOS、安卓和 Windows Phone 各自的设计规范越来越正式，特别是导航方面。

所以，了解这些操作系统的指导原则比以往更加重要，而更关键的是，你应该真正熟悉你的客户目前时刻使用的设备。我强烈建议你至少花六周的时间使用每一个你为之设计应用的设备。这样，当你挽起袖子开始工作时，你自己的经验加上本书提供的设计模式会让你充满信心地设计出优美易用的应用。

## 本书读者

本书面向创造移动应用的产品经理、设计师和开发者。随着公司不断定义和改善他们的移动战

略，找到最佳设计的案例是一大挑战，特别是对多操作系统来说。无论你是负责设计一款简单的iPhone 应用，还是为市面上每一个流行的操作系统做设计，这些设计模式都会为常见的设计挑战提供解决方案。

## Safari® Books Online




Safari Books Online（http://www.safaribooksonline.com）是应运而生的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。技术专家、软件开发人员、Web 设计师、商务人士和创意专家等，在开展调研、解决问题、学习和认证培训时，都将Safari Books Online 视作获取资料的首选渠道。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

## 联系我们

请把对本书的评价和问题发给出版社。

美国：

O’Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O’Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://oreil.ly/mobile-design-2e

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestio ns@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

http://www.oreilly.com

我们在 Facebook 的地址如下：

http://facebook.com/oreilly

请关注我们的 Twitter 动态：

http://twitter.com/oreillymedia

我们的 YouTube 视频地址如下：

http://www.youtube.com/oreillymedia

## 致谢

Rich Malley 帮助我完成了此书，我欠他一个人情；我希望这一次是我们合作的开端。非常感谢 Cathlin McCullough 帮我组织整理了社交模式一章。感谢 Suze Kemper 在截稿日期临近时给予我莫大的帮助。万分感谢 Ivan Bachev 再一次帮我准备了全部 1000 张图片。

感谢 Alissa Briggs、Greg Nudelman 和 Eli Holder 分享他们的故事，感谢 Aaron Jansinski 提供的模式图解。

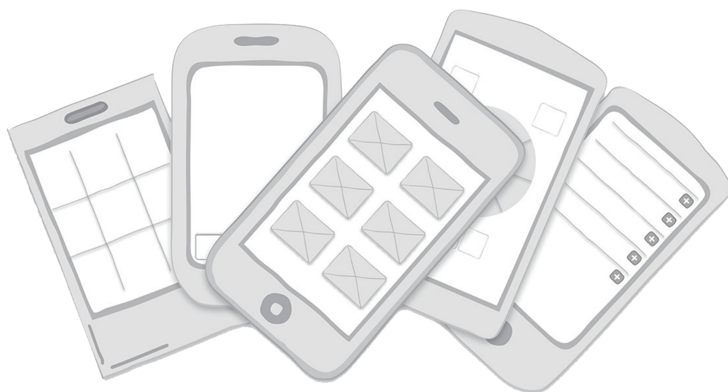
我还想感谢所有富有创造力的敬业的移动应用开发和设计团队。本书展示了其中很多了不起的应用，我为此感到非常荣幸，同时对他们付出的艰苦努力深表感激。

这是我为 O'Reilly Media 写的第三本书，能够与 Mary Treseler 以及她的团队再次合作，我感到非常愉快。

最后，向我们家庭的新成员 Marlena Elizabeth Ann 致以特别的感谢，是她给了我动力，让我在她出生之前（还有整整 12 个小时）完成本书。

# 第 1 章

# 导航



- 主导航模式——全局导航

  跳板式（Springboard）、列表菜单式（List Menu）、仪表式（Dashboard）、陈列馆式（Gallery）、选项卡菜单式（Tab Menu）、隐喻式（Skeuomorphic）

- 主导航模式——瞬时导航

  侧边抽屉式（Side Drawer）、下拉菜单式（Toggle Menu）、转盘菜单式（Pie Menu）

- 次级导航模式

  翻页式（Page Swiping）、滚动选项卡式（Scrolling Tabs）、伸缩面板式（Expand/Collapse Panel）

我喜欢查看移动应用市场上的用户评价，以便更深入地理解人们如何使用应用。移动应用市场的评分系统提供了极有价值的反馈信息，这种反馈在网页和桌面应用领域是不存在的。这些信息很好地反映了用户的偏

好和期望。

总体上来说，大多数 4 星和 5 星评价描述得并不具体，基本上就是“这个应用太棒了，界面好看，用起来顺手”，诸如此类。但 1 星和 2 星评价就很有看头了，这样的评价往往更为真实地反映了用户在使用过程中遇到的问题。最常见的抱怨如下：

- 软件崩溃
- 关键功能缺失（比如同步、筛选、关联账号）
- 界面设计混乱
- 导航欠佳（比如无法返回，找不到要找的东西）

设计模式并不能解决前两个问题，因为这两个问题都需要进行用户测试和设备测试，但肯定可以解决后两个问题。遵循常见的导航设计模式可以确保用户能够在你的应用中找到并使用其中有价值的功能。

良好的导航与优秀的设计如出一辙，都是低调内敛的。好的导航设计使得应用软件简单易用，无论是浏览图片，还是申请购车贷款，用户都可以轻松完成。

## 主导航模式——全局导航

接下来，我们要介绍的第一组模式用于主导航，比如从一个主类别切换到另一个主类别，就像桌面应用程序的一级菜单。自本书第 1 版出版以来，主导航已经演化成了两种不同的类型：全局（persistent）导航和瞬时（transient）导航。

全局导航包括列表菜单式、选项卡菜单式等简单的菜单结构。只要打开有全局导航的应用，主导航的各个选项立刻一目了然。

而瞬时导航必须通过点击或手势才会清楚地显示出来。由于智能手机的屏幕尺寸十分有限，这让移动应用设计师们不得不思考如何“打破（屏幕的）框架”，于是这种导航模式渐渐流行起来。

对我来说，经典的 9 点谜题游戏完美阐释了移动应用导航设计模式的思维变化。试试看：用 4 条或更少的直线将全部 9 个圆点连接起来，要一笔完成，且各条线之间不能重叠。

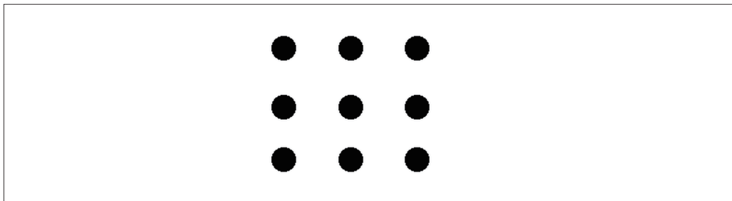


图 1-1
9 点谜题

你会怎么做呢？你可能已经发现，解决这个谜题的唯一办法是打破虚拟的边界。在移动领域，这一思想叫作“突破画布”（off-canvas）。

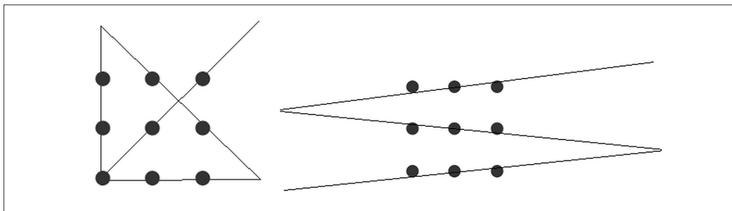


图 1-2
9 点谜题的两种解法

这种突破画布思维启发了侧边抽屉设计模式，这种模式成为现在 iOS 和安卓应用上最受欢迎的主导航模式之一。

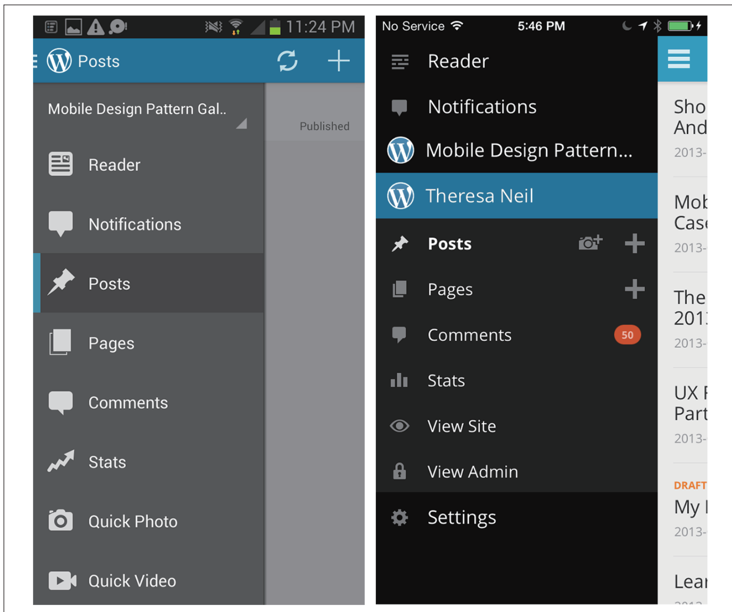



图 1-3
iOS 版和安卓版 WordPress：侧边抽屉设计体现了“突破画布”的思想

“打破虚拟边界”的思想也在很大程度上影响了 Windows Phone 8 和 Ubuntu Touch（一个新的开源移动操作系统）。

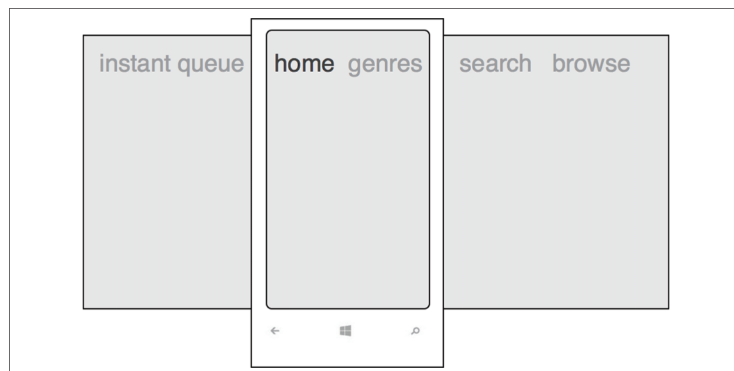



图 1-4
Windows Phone 的全景控件设计

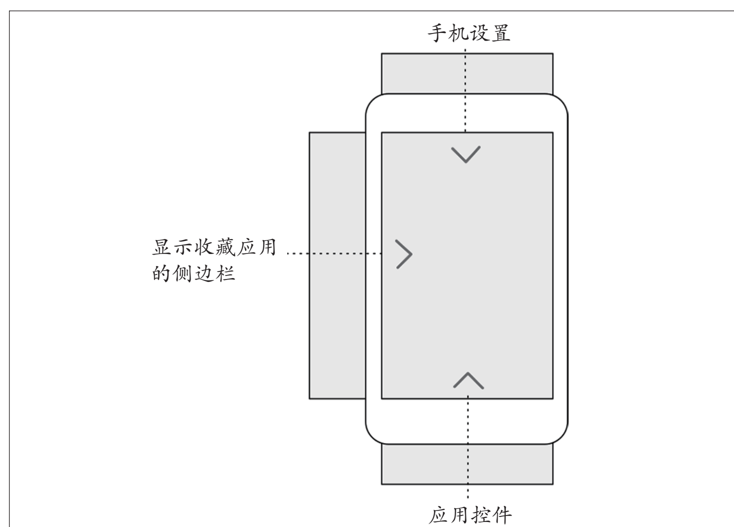



图 1-5
在 Ubuntu 移动系统中，可以通过滑动屏幕边缘来调出设置选项和菜单，这样可以用整个屏幕来展示应用内容

设计师们的设计思维从此发生了巨大转变，他们开始将内容层叠起来，而不是将 UI 统统放到一个平面上去。通过 3D 图层面板和手势交互创造极佳用户体验的一个很好的例子，是 Twitter 早期的 iPad 应用：左侧面板是菜单栏，中间面板是内容列表，右侧面板显示详细内容。点击中间部分的任意一条内容时，左侧菜单将收起，同时在右侧面板显示内容预览。点击右侧面板，右侧面板将展开，大小占到全屏的 70% 左右。




**图 1-6**

iPad 版 Twitter 的早期版本：图层面板和手势交互利用了移动平台的特性

在决定使用全局导航还是瞬时导航时，你可以自问以下几个问题：

- 你的应用层次是否“扁平”？应用的菜单分类是同级的还是有优先级的？主要类别是否只有少数几个，即 3 到 5 个？
- 你的用户是否需要菜单一直处于可见状态以便快速访问？
- 菜单分类有无状态指示，比如未读邮件数？

如果对于以上问题，你的回答中至少有一个“是”，那么你可能最好使用全局导航。现在就让我们看看这些设计模式吧。

## 跳板式

跳板式设计模式，也称为启动面板（Launchpad），是 2011 年最流行的导航模式。该模式指的是一个带有菜单选项的登录界面，而菜单选项就是进入各个应用的起点。

跳板式模式风行一时的原因之一是跨平台。当时，我们很多人都还在摸索没有操作系统倾向的、具有兼容性并可重复利用的设计。跳板式流行起来的另外一个原因是，这种模式一屏最多可以放置 9 个选项（3×3 格）。相比 iOS 和安卓系统标签栏上 3 到 5 个选项卡的限制，这种设计大大增加了选项容量。而且如果再在页面上加入翻页指示（底部的小圆点），那么设计师还可以提供更多的菜单选项。

图 1-7

iOS 版 Trulia 和安卓版 Gowalla

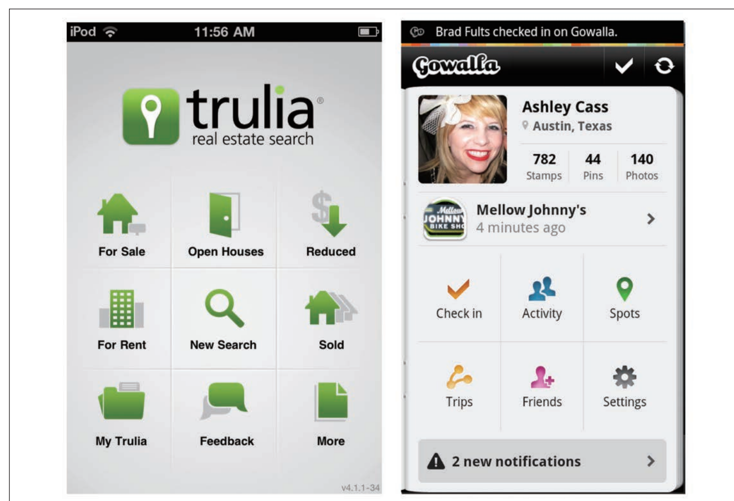





图 1-8
iOS 版 Facebook 和安卓版 LinkedIn：2011 年版中的跳板式设计

跳板式设计模式的主要缺点是全部选项都被扁平化，没有任何优先级。而开启抽屉式模式时代的，是 Aza Raskin 设计的火狐浏览器移动版（http://www.azarask.in/blog/post/firefox-mobile-concept-video）。2011 年，抽屉式被 Path 采用。这种模式比选项栏容纳的选项更多，而且可以通过对这些选项进行逻辑分组来表达重要性或优先级。我们稍后会在本章进行探讨。

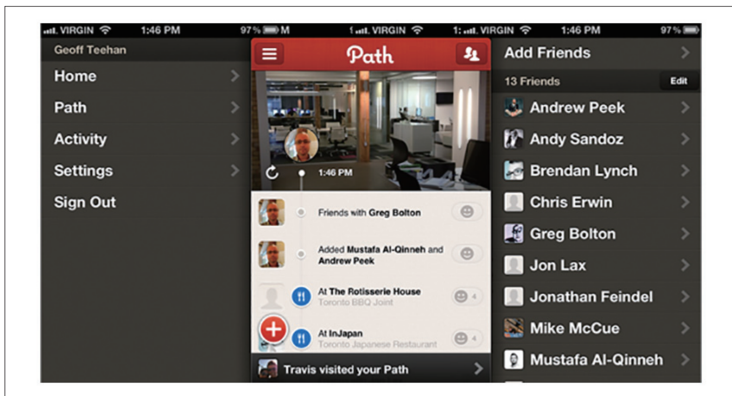
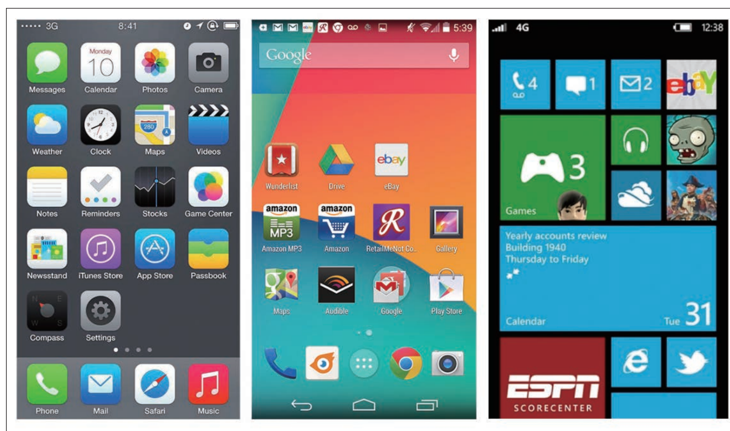



图 1-9
iOS 版 Path（2011 年 11 月）：进入抽屉式设计时代

然而，跳板式设计模式并没有消亡。安卓、iOS 和 Windows Phone 均在操作系统层级延续了这一导航模式。

图 1-10

iOS 7、安卓 4.4（奇巧）和 Windows Phone 8 均在操作系统层面使用了跳板式导航模式




至今还有很多应用仍在使用传统的跳板式设计。LearnVest、BBC Radio 和 Vimeo 分别采用了标准的 4 宫格、6 宫格和 9 宫格布局。

图 1-11

iOS 版 LearnVest、Windows Phone 版 BBC Radio 和安卓版 Vimeo：应用上的传统跳板式设计不死，而且看上去不错




Orbitz 和 EasyJet 采用不同的图标设计和栅格布局，从视觉上展现出了菜单的优先级。

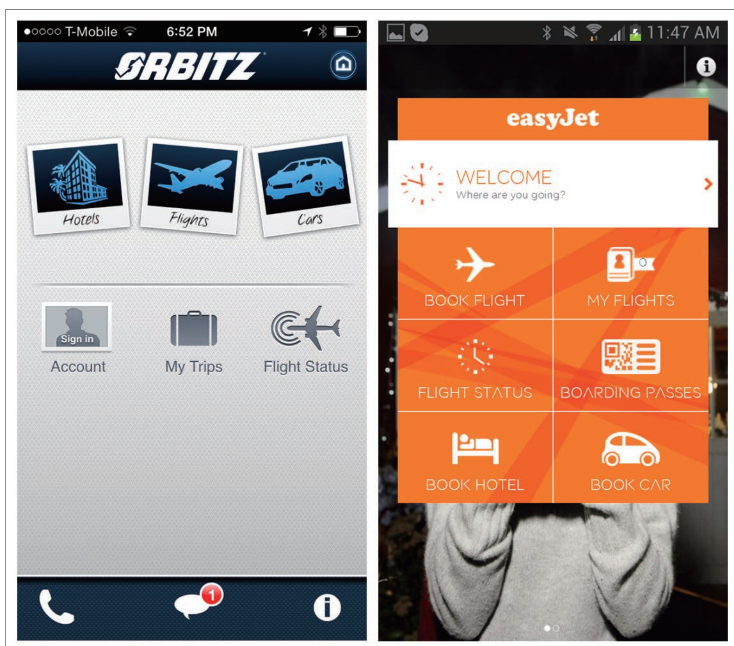


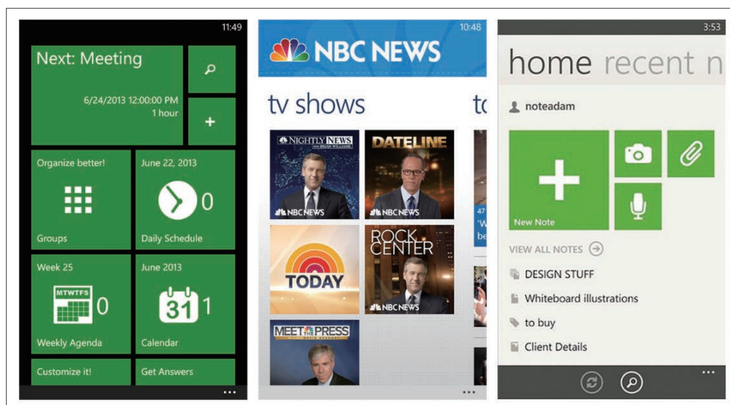

图 1-12
iOS 版 Orbitz 和安卓版 EasyJet：图标和布局暗示层次关系

Windows Phone 的磁贴模式（tiles）将跳板式设计发挥到了极致。磁贴有动态的，也有静态的，拥有三种不同的尺寸。动态磁贴可以显示未接来电数、下次约会安排情况或是最近联系人头像这样的动态信息。要了解更多关于 tiles 的信息，请阅读 Windows 设计指南：http://msdn.microsoft.com/library/windows/apps/jj662929(v=vs.105).aspx。

Windows Phone 的磁贴可以用来做主导航，或者作为一个二级导航模式搭配全局控制来使用。下面三个应用的例子显示了 Windows Phone 磁贴的多功能性。CalendarPro 利用动态磁贴做主导航设计，NBC News 则设计了子导航栏，而 Evernote 利用静态磁贴做了子导航栏的设计。

图 1-13

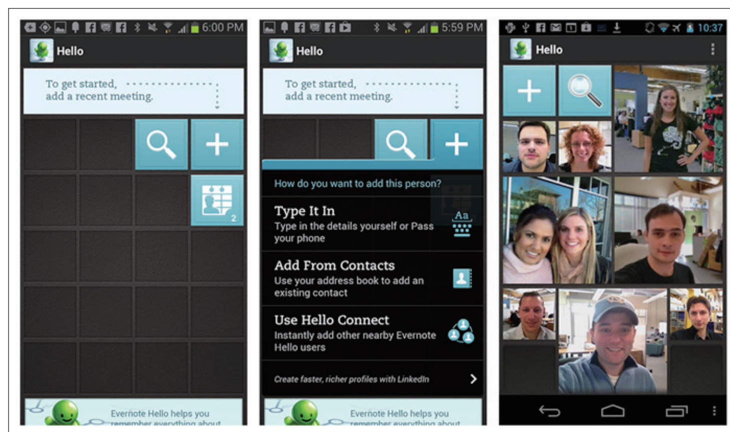
Windows Phone 版 CalendarPro、NBC News 和 Evernote 展示了各种磁贴的实现




安卓和 iOS 版的 Evernote Hello 采用了基于磁贴模式的跳板式导航设计。用户可以通过先添加人物再添加会议来自定义用户界面。

图 1-14

安卓版 Evernote Hello：经磁贴启发的可定制的跳板式设计




## 卡片式

2010 年到 2011 年使用过 Palm 手机的人可能都熟悉卡片式导航。卡片式导航的原型是扑克牌，模仿了扑克牌中常见的切牌、洗牌、弃牌、翻牌等手法。

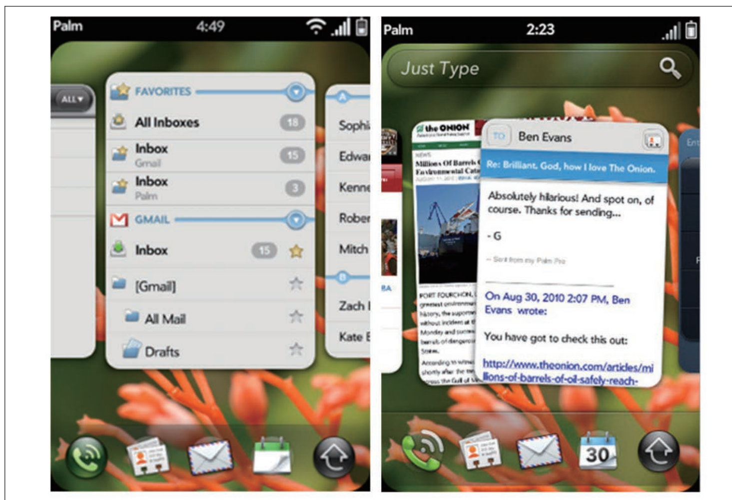



图 1-15

2010 年到 2011 年左右的 Palm webOS 系统：应用像纸牌一样，呈扇形展开

这种模式随着 Google Now 应用的发布而重获新生，Google Now 用一连串信息卡堆叠的形式展示一系列进入应用的切入点，或是阅读内容的快捷操作。

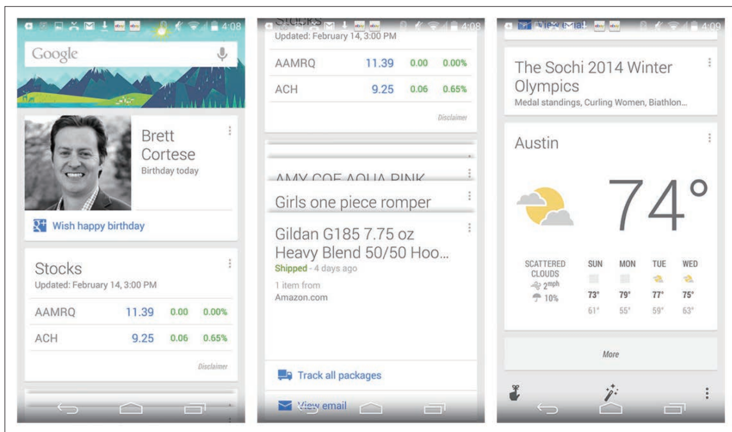


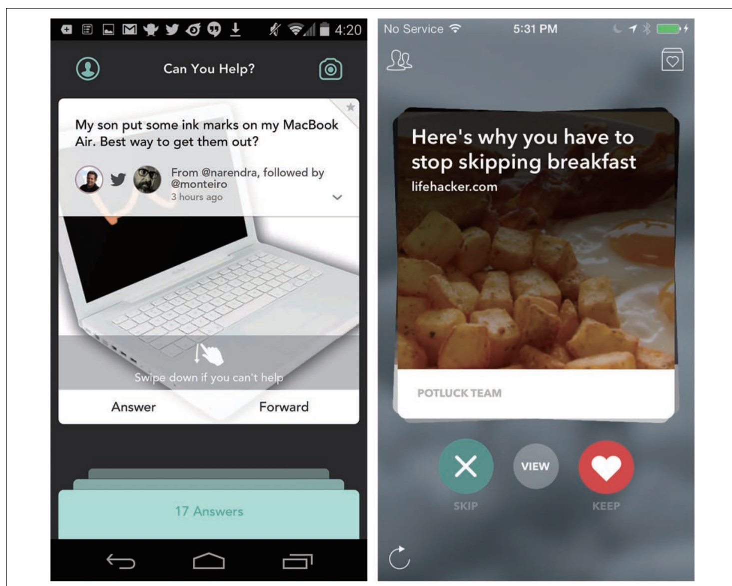

图 1-16

iOS 版和安卓版 Google Now：卡片式主导航

Jelly 和 Potluck 应用也一脉相承地采用了卡片式设计作为导航和内容交互的主要模式。在 Jelly 里，向下滑动卡片可以将其从屏幕上移走，这意味着你不能回答别人的问题，与此同时新的卡片取而代之。在 Potluck 中，向左滑动一叠卡片的最上面一张可以跳过这个故事，向右滑动会将这张卡片移动到“保存的卡片组”中。

图 1-17

iOS 版 Jelly（http://www.youtube.com/watch?v=bCDB_TrAhSY）和 Potluck（http://www.youtube.com/watch?v=pcfNFuvvdrA）




［小贴士］

卡片式设计模式给出了一种展示内容的优雅方案。真正的卡片式模式还应该包括模拟切牌、换牌、翻牌等的交互。

Facebook 和 Pinterest 利用了卡片式的视觉风格，但与前面提到的例子相比缺少了手势交互。这让二者更像卡片风格的列表元素，而非真正的卡片式设计。

要想了解误用卡片式设计模式的例子，请参阅第 11 章“标新立异”一节中关于阿拉斯加航空（Alaska Airlines）应用的讨论。

## 列表菜单式

列表菜单式中的每一个列表项都是进入该应用各项功能的入口，这一点与跳板式类似，并且模块之间的切换需要返回到列表主页。苹果的设计规范（https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/Navigation.html#//apple_ref/doc/uid/TP40006556-CH53-SW1）将其称为层级导航（hierarchal navigation）：

> 在层级式的应用中，用户通过在每个页面选择一次进行导航，直至到达目标位置。要到达另一个位置，用户必须原路返回几步（或者从头开始），然后做出不同的选择。“设置”和“邮件”是使用层级式结构应用的典范。

Kayak、Day One 和 AroundMe 应用的页面展示出各种不同的列表菜单样式。

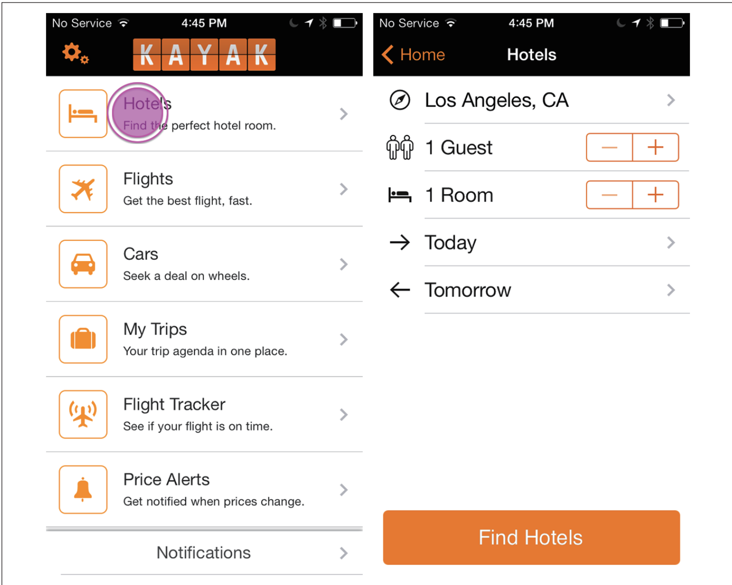



图 1-18

iOS 版 Kayak：点击“首页”（Home，见右图）返回列表菜单

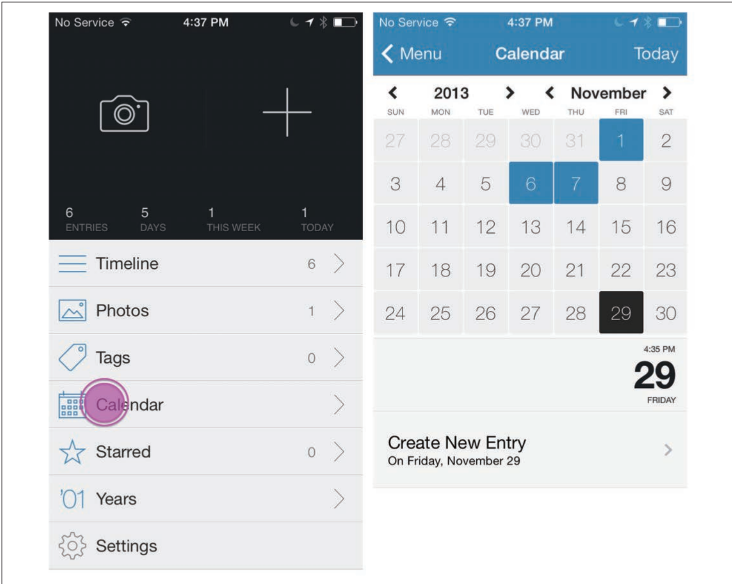
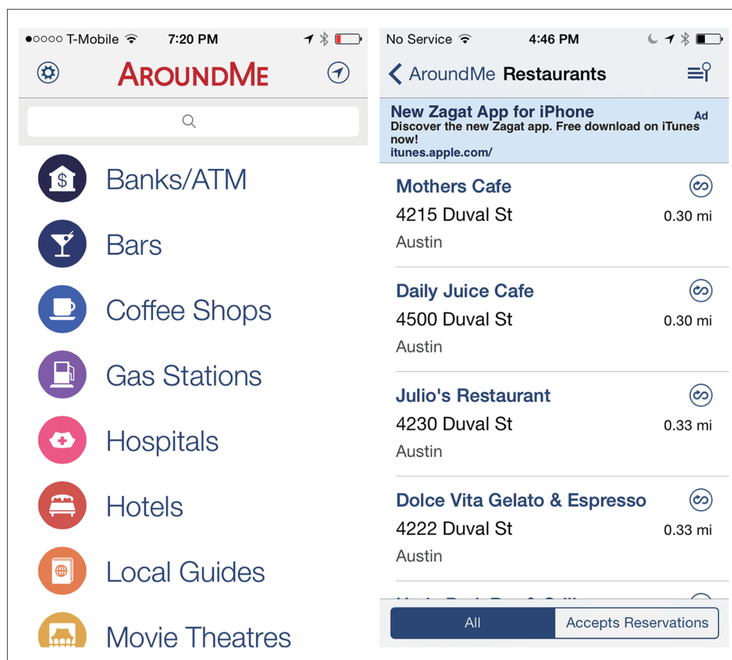



图 1-19

iOS 版 Day One：列表菜单式的主导航

图 1-20

iOS 版 AroundMe：返回按钮文字也许叫作“首页”（Home）或“菜单”（Menu）会更好一些




安卓中的列表菜单式导航大同小异，只是表达这种模式的层级结构的返回按钮（Back button）被称为向上按钮（Up button）。关于这一点，安卓设计指南中写道：

> 向上按钮用来在应用的各个页面之间根据层级关系进行导航。举例来说，如果页面 A 显示一个条目列表，点击其中一项进入页面 B（显示该项的详情），那么页面 B 应该有返回页面 A 的向上按钮。如果某个页面已经是应用的顶级页（即首页），那就不需要向上按钮。

eBay 应用就是这样一个例子：向上按钮由 eBay 图标前加左箭头组成。请注意，多数用户的预期是左箭头和 logo 或图标同时可点。

［小贴士］

在设计层级结构的应用导航时可以考虑列表菜单式。这种模式对于列表项名称较长的菜单和列表项既要有名称又要有描述的菜单都同样适用。实现这种导航模式要遵循不同操作系统的设计规范。

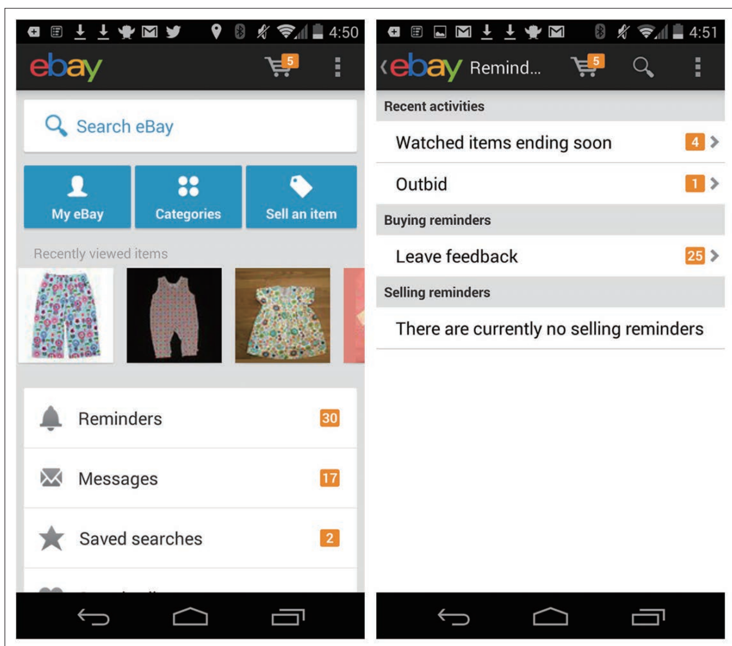



图 1-21
安卓版 eBay：左箭头图标是向上按钮

## 仪表盘式

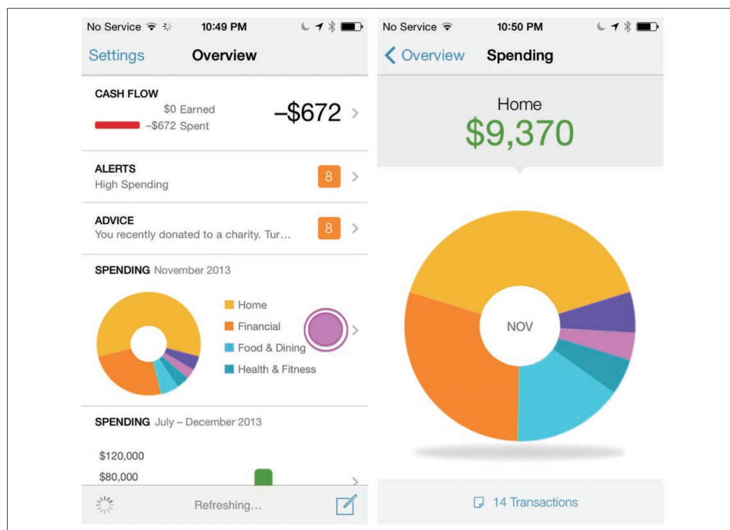
仪表盘式与跳板式和列表菜单式相似。

设计优秀的仪表盘让用户一眼就能看到他最需要的信息概览，而不用转到另一个页面。

当你逐级深入设计各界面时，要提供能够返回到仪表盘的导航，这一点与列表菜单式和跳板式的导航设计规则相同。有关仪表盘式设计的更多内容请参见第 6 章。

图 1-22

iOS 版 Mint：仪表盘将数据作为各选项的入口




［小贴士］

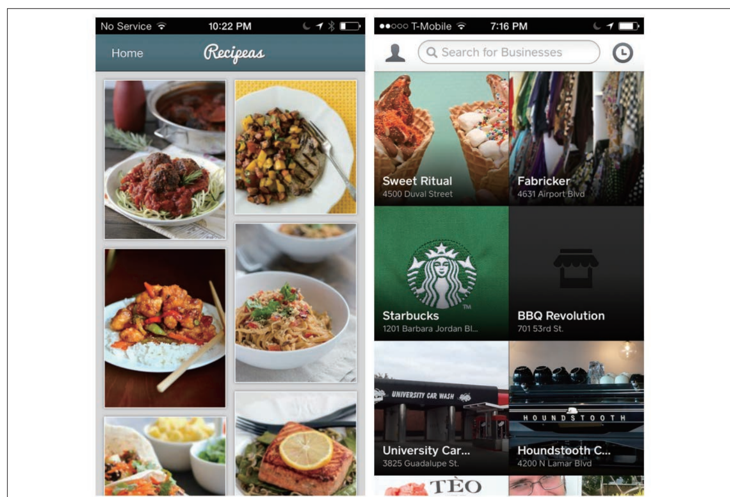
在需要使用关键指标或数据作为应用入口的时候，可以采用仪表盘式。但不要载入过多信息，对于要展示的关键指标或数据需经过仔细研究再做决定。

## 陈列馆式

陈列馆式设计可用来呈现实时内容，比如新闻、菜谱或照片，可以采用网格布局（比如 Recipeas 和 Square Wallet 应用）或轮盘布局（比如 LinkedIn Pulse 和 BBC News 应用），还可以采用幻灯片模式进行展示。

图 1-23

iOS 版 Recipeas 和 Square Wallet：陈列馆式设计展示了彼此独立、无层级关系的内容项



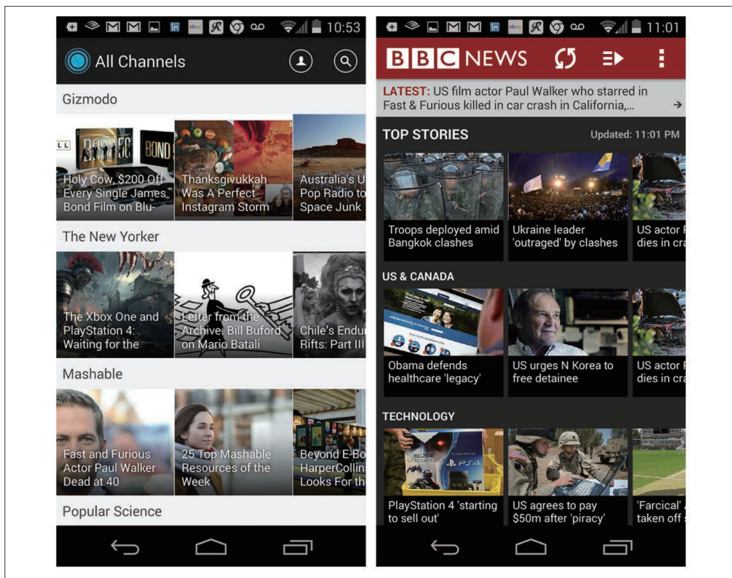



图 1-24
安 卓 版 LinkedIn Pulse 和 BBC News：字幕式文本比浮层式文本更清晰易读

［小贴士］

陈列馆式设计模式最适合呈现经常更新的、视觉效果直观的无层级内容。

请留意一下，快速浏览 BBC News 的内容要比 LinkedIn Pulse 轻松，这是因为前者的文章标题位于图片下方，而不是覆盖在图片上。

## 选项卡菜单式

就选项卡式设计而言，安卓、iOS 和 Windows Phone 都有各自特定的术语和设计规范。在此我会把这些规范一一列出来，这是因为，即使你决定在设计迭代中抛开这些规范，理解它们的原则也非常重要。

### iOS

苹果自第 1 版 iOS 系统发布以来，就建议我们把标签栏作为扁平式应用的导航（https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/Navigation.html）：

> 在扁平式信息结构的应用中，因为所有的主要类别都可以从主页面进入，所以用户可以直接从一个主要类别切换到另一个。音乐和应用商店便是运用扁平式结构的范例。

有意思的是，Facebook 在采用侧边抽屉式导航长达两年之后，最近又回归到了选项卡式设计。你可以浏览网页 http://tcrn.ch/1dZFlUF，阅读更多关于 Facebook 用户测试过程和结果的信息。

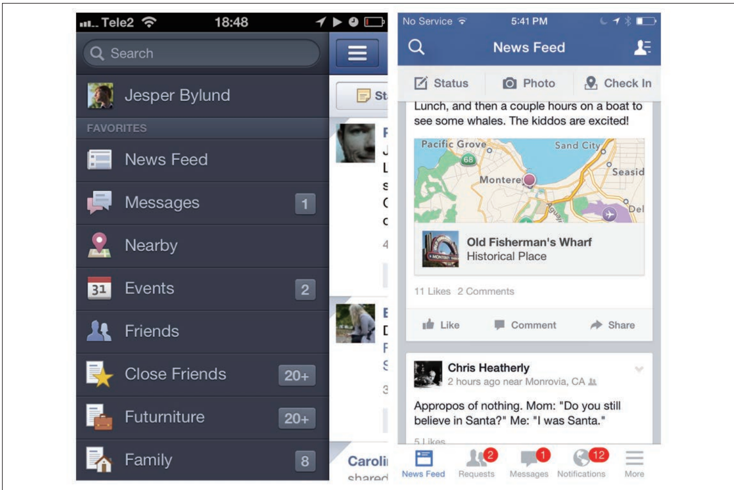



图 1-25

iOS 版 Facebook 的新旧版本：在千万级用户批量测试中，选项卡式（右图）完胜侧边抽屉式（左图）和其他类型的导航模式

iOS 系统中的标签栏限制在 5 个选项卡以内。如果应用的主要类别超过 5 个，那么你可以把“更多”放在最右侧的第五个选项卡上。

理解 iOS 中的标签栏和工具栏之间的区别十分重要。标签栏的作用是在应用的主要类别之间进行切换，而工具栏上显示的是工具，或是指向特定页面的可执行操作。

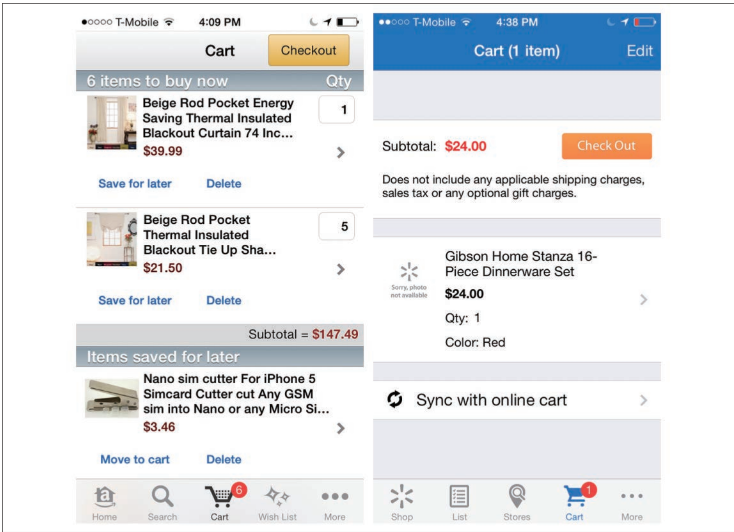



图 1-26

iOS 版亚马逊和沃尔玛：标签栏的应用

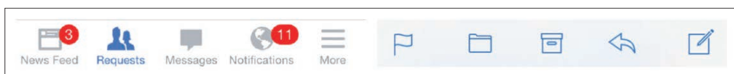



图 1-27
标签栏（左图）上有选项卡菜单；工具栏（右图）上有可操作的工具

有些应用，例如 Instagram 和 RunKeeper，主要针对进行单一操作的用户（如拍摄照片或者开始跑步），所以他们把行为召唤（calls to action，一项优先的行动）放置在标签栏上。

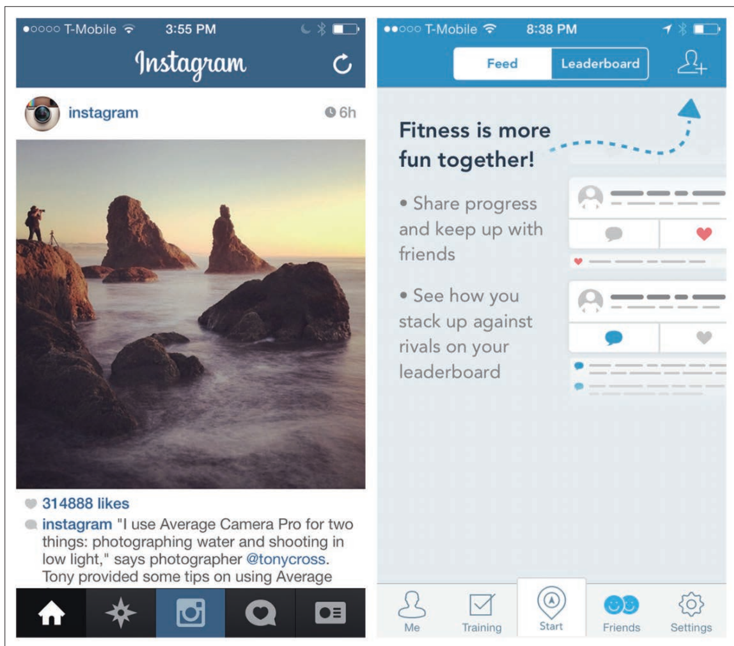
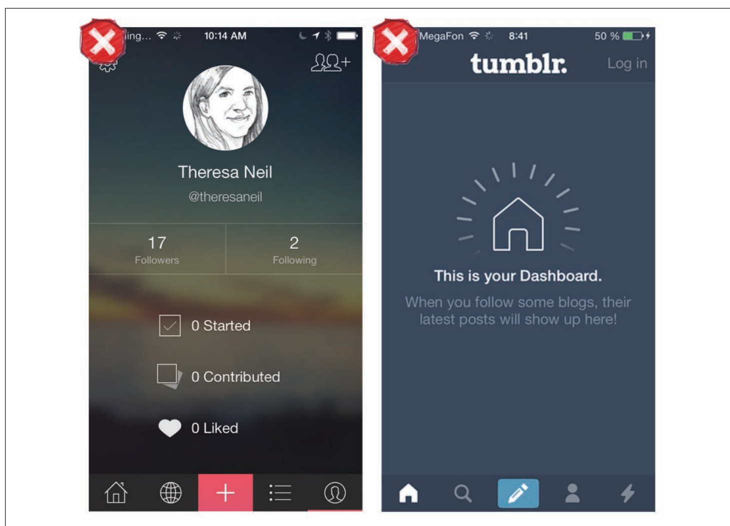



图 1-28
iOS 版 Instagram 和 RunKeeper：标签栏上的行为召唤按钮

如果你要设计这种标签栏，就要确保突出选中的选项卡。比如，在 Everlapse 和 Tumblr 中，你就很难分辨出选中的是哪一个选项卡，因为选中项过于黯淡，完全被主要操作按钮的视觉效果所掩盖。

图 1-29

iOS 版 Everlapse 和 Tumblr：突出的操作按钮掩盖了选项卡菜单的位置指示




## 安卓

安卓系统为顶级导航或称主导航提供了三种不同的选项卡菜单模式：固定选项卡（Fixed Tabs）、下拉菜单（Spinners）和导航抽屉（Navigation Drawers）。以下链接可以查看安卓设计指南中关于固定选项卡的部分：http://developer.android.com/design/patterns/app-structure.html。

> 固定选项卡同时显示多个顶级内容视图，可以方便地在不同视图间进行浏览和切换。固定选项卡会一直显示在屏幕上，不能像滚动选项卡（Scrolling Tabs）那样移动。固定选项卡应该始终允许用户通过在内容区向左滑或向右滑，来切换不同的界面。
>
> 在以下场景中使用选项卡：
>
> - 预计你的用户会经常切换视图。
> - 顶级内容视图有限，最多 3 个。
> - 你想让用户清楚地知道有可供选择的视图。

Path 使用图标作为固定选项卡，而 Quora 把 4 个文字选项卡塞进了有限的空间里。设计师往往错误地在主导航中使用滚动选项卡，而实际上应该使用下拉菜单或导航抽屉。

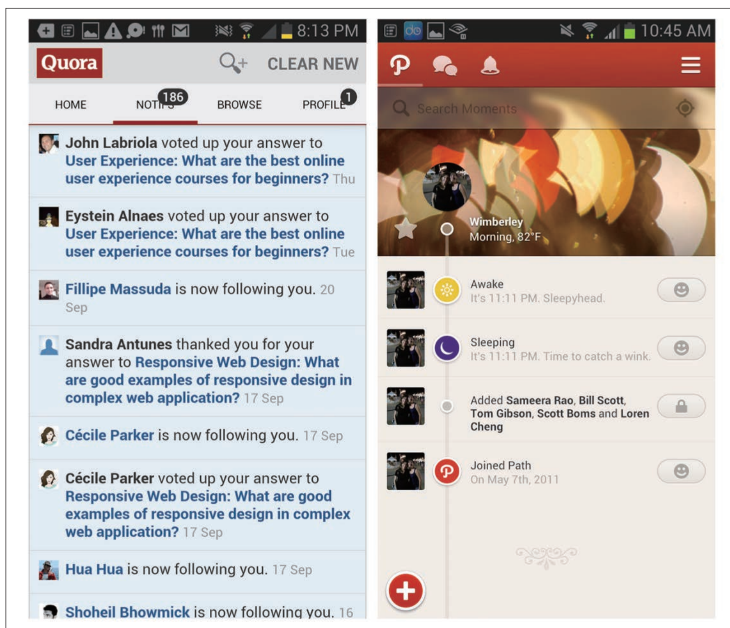



图 1-30

安卓版 Quora 把 4 个文字选项卡塞进了有限的空间里；安卓版 Path 使用图标作为固定选项卡

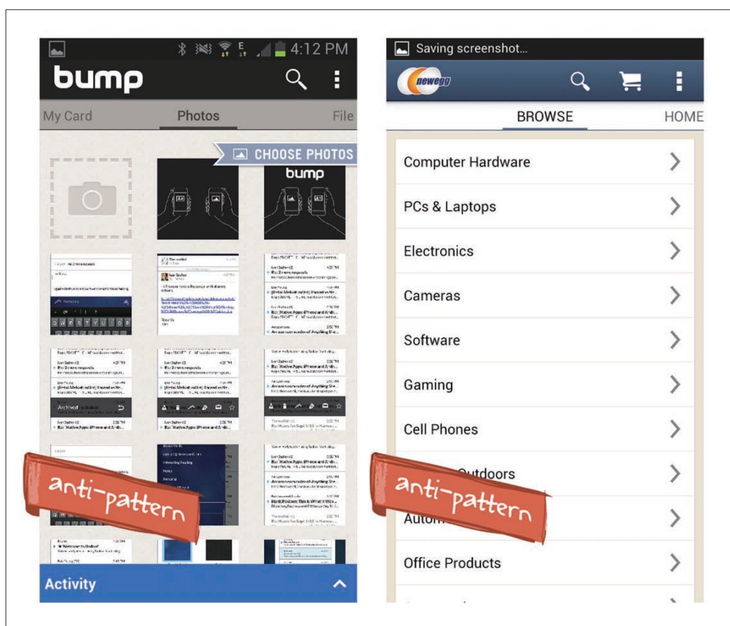



图 1-31

安卓版 Bump 和 Newegg；错误地将滚动选项卡作为主导航

## Windows Phone

Windows Phone 系统中的选项卡菜单被称为应用选项卡（App Tabs），你可以通过水平滚动来浏览延伸至屏幕之外的选项卡页面，这一模式是通过透视控件（Pivot control）实现的。Windows Phone 设计指南中（http://msdn.microsoft.com/library/windows/apps/hh202890(v=vs.105).aspx）建议道：

> 你可使用透视控件（http://msdn.microsoft.com/library/windows/apps/microsoft.phone.controls.pivot(v=vs.105).aspx）实现“应用选项卡”模式。该控件允许用户在每个页面（称为透视项目[1]）之间向左右水平导航。

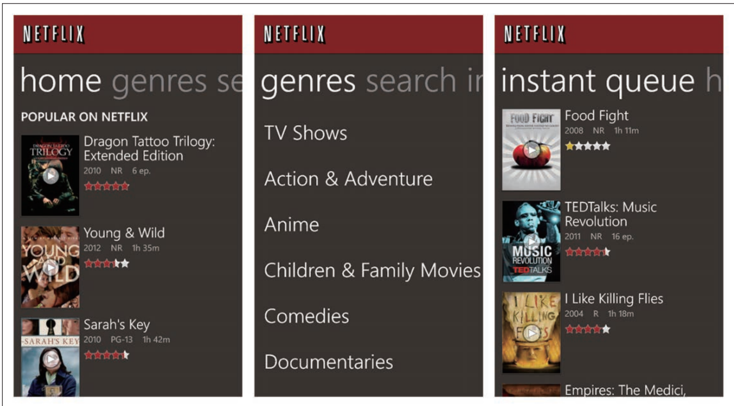



图 1-32

Windows Phone 版 Netflix：透视控件实现应用选项卡模式

## 新生模式

无论是桌面端还是移动端网页，都有一种新的设计趋势，就是在用户滚动屏幕或向下滑动页面内容时把网站头部隐藏或是收缩起来。

这种折叠选项卡式（Retracting Tab）设计也出现在了原生应用中。Pinterest 就在用户向下滑动浏览内容时把工具栏收起来，在向上滑动时工具栏又重新显示出来。iOS 版的 Luvocracy 和 Polar 也用到了这种效果。

注 1：透视项目译法，见 http://msdn.microsoft.com/zh-cn/library/windows/apps/dn596104.aspx。（译者注）

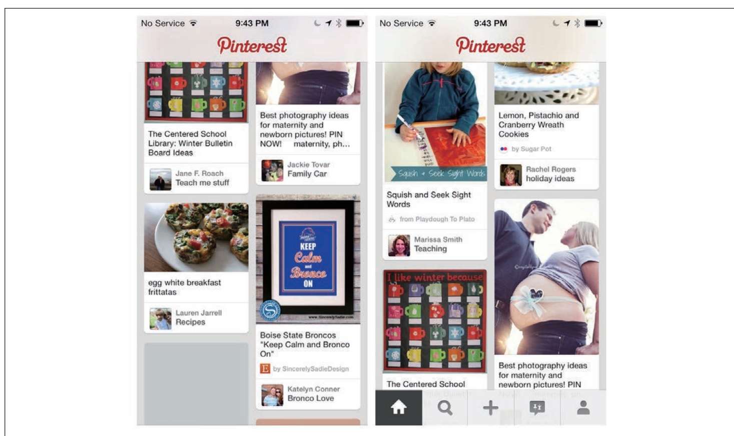



图 1-33
iOS 版 Pinterest：向下滑动隐藏工具栏，向上滑动显示工具栏（http://www.youtube.com/watch?v=joaaJgvTN28）

可配置选项卡式（Configurable Tabs）是标准选项卡菜单式的另一个变种。Frequency 的设计模仿了所有主要桌面网页浏览器上的标签页效果。添加一个频道就增加一个新的选项卡。如果屏幕上已经有太多选项卡，头部就可以滚动。用户只需长按一个选项卡，再拖动到目标位置，就可以调整选项卡的顺序。

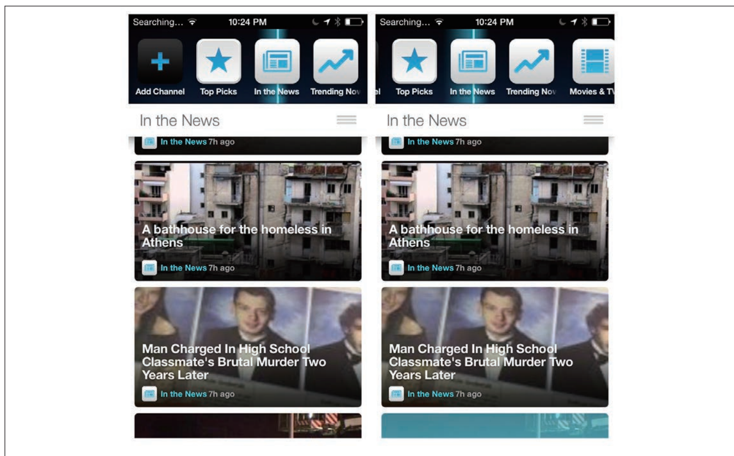
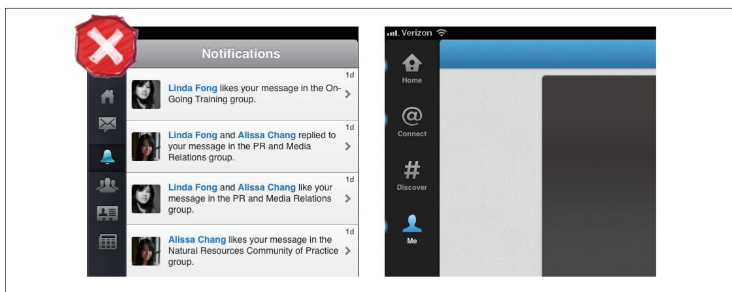



图 1-34
iOS 版 Frequency：新生模式之一的可配置选项卡式，效果类似于网页浏览器的标签页

原生应用中的侧边栏也像网页应用中的侧边栏一样愈发流行起来。Twitter 在其 iPad 应用中，用清晰的文字标识主界面的侧边栏导航。Yammer 的选项卡数量几乎是 Twitter 的两倍，但是没有文字标识，这让导航用起来有点困难。

图 1-35
iPad 版 Twitter 的选项卡文字标识清晰，iPad 版 Yammer 则没有文字标识

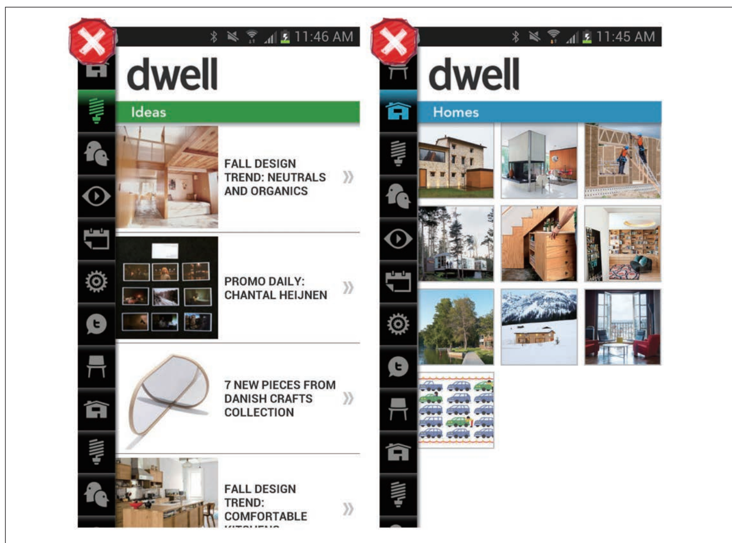



侧边栏不太可能被广泛用作智能手机的全局导航模式，这有两个原因：

- 大多数人都是在竖屏模式下使用智能手机，而侧边栏需要占用相当多的横向空间。
- 因为空间有限，所以文字表示往往被去掉，这就降低了应用的可用性。

Dwell 是一个能起到警示作用的例子。这个应用看上去挺不错，但你不得不点击每个图标去看看各个主要类别究竟是什么——这真是一个“肉之迷导航”（mystery meat navigation，http://www.webpagesthatsuck.com/mysterymeatnavigation.html）的经典案例。

图 1-36
安卓版 Dwell：肉之迷式的侧边栏导航




［小贴士］
要了解并遵守不同操作系统对选项卡菜单式的设计规范。用选中和未选中选项卡间的视觉差异明确指示出用户所处的位置。

## 隐喻式

隐喻式（或拟物式）的特征是用页面模仿应用的隐喻对象。尽管现在视觉设计的审美倾向于扁平化设计，但一些应用仍然能够通过模仿真实世界的物体和工具提升可用性。

隐喻式导航主要用在游戏设计中，比如《狙击手：幽灵战士 2》（*Sniper: Ghost Warrior 2*），但在一些非游戏类型的应用中也能看到，如混音应用软件 Cross DJ、照片应用软件 Hipstamatic，以及旅行类应用软件 FlightBoard。



图 1-37
安卓版和 iOS 版《狙击手：幽灵战士 2》：拟物风格在游戏导航中十分普遍




图 1-38
iOS 版 Cross DJ：拟物设计也见于一些非游戏类的应用




图 1-39
iOS 版 Hipstamatic：对真实物体的模拟

还有其他一些例子，其中包括 iOS 7 版乐顺备忘录（Awesome Note）和苹果的报刊杂志（Newsstand）的新设计。这两个应用的设计能让人立刻识别出来，并且非常直观，因为我们可以看出它是在模仿文件夹和书架。

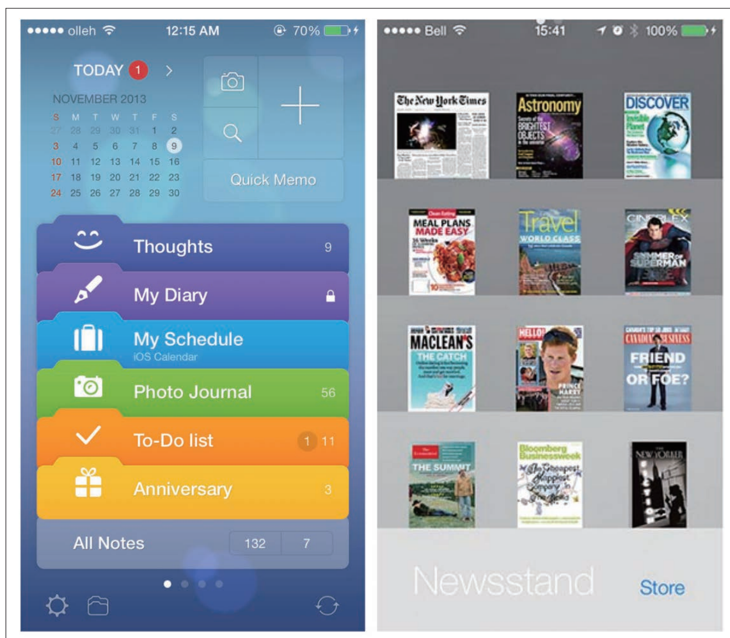



图 1-40
iOS 版 Awesome Note 和 Newsstand：拟物风格使得导航更直观

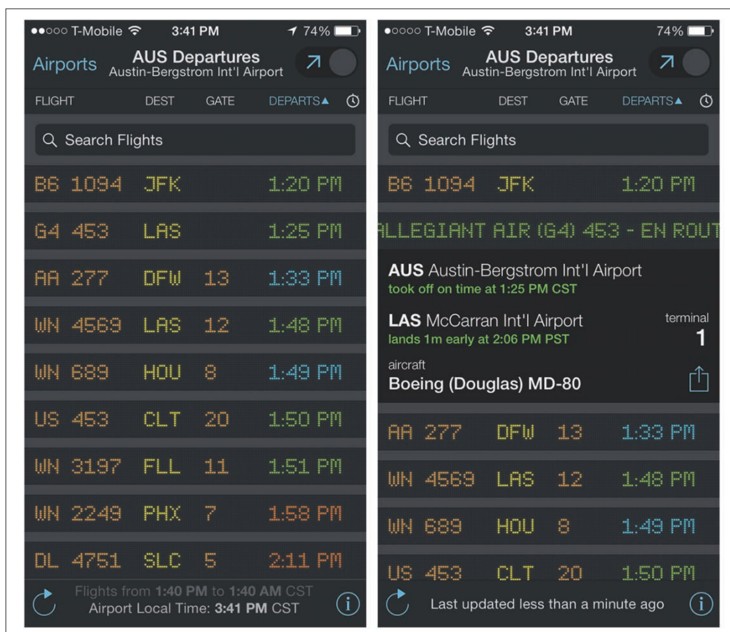



图 1-41
iOS 版 FlightBoard：设计成机场里的航班信息显示系统的模样

［小贴士］

一定要谨慎地使用隐喻式导航，蹩脚的模仿很可能造成第 11 章中讨论的标新立异反模式（Novel Notion anti-Pattern）。

当然，延伸到电子领域的任何隐喻都是有限的。比如说，早期版本的 iBook 就一直让人有些迷惑：一些用户认为电子书架的大小暗示了它能装下的电子书的数量，然而实际上并非如此。

## 主导航模式——瞬时导航

“瞬时”一词意为短时间停留，它精确地表达了下面这种导航菜单的作用。这种导航菜单只有在我们需要的时候才显示出来，在我们做出选择后，它会再次消失。这里我们要介绍三种瞬时导航：侧边抽屉式、下拉菜单式和转盘菜单式。

### 侧边抽屉式

侧边抽屉式有两种风格。第一种是*浮层*（overlay），通过轻滑或点击的手势打开抽屉，抽屉部分遮挡或覆盖原来页面的内容，例如 RetailMeNot。第二种风格是*嵌入层*（inlay），通过轻滑、平移或点击打开抽屉，把原先的页面内容部分推出屏幕外，例如 Path。

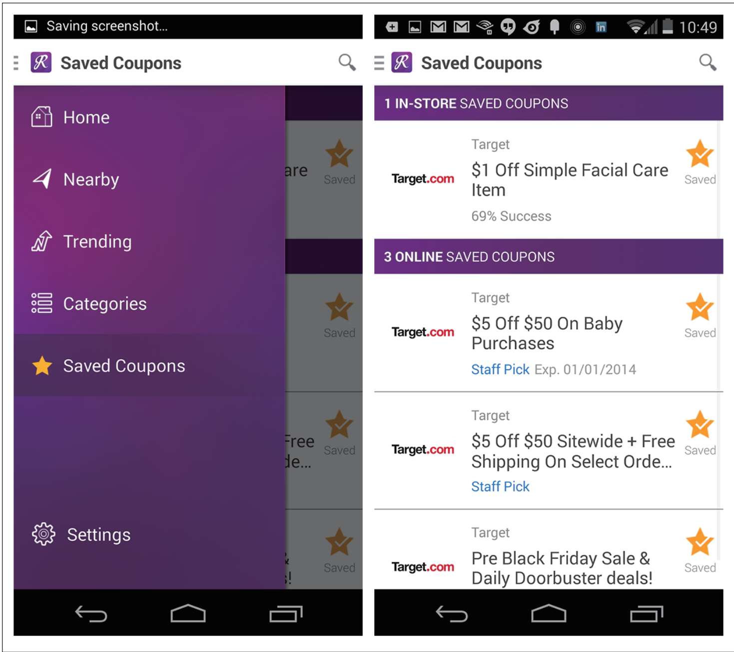



图 1-42

安卓版 RetailMeNot：点击导航图标或在屏幕边缘轻滑，以显示侧边抽屉浮层，浮层部分覆盖主页内容

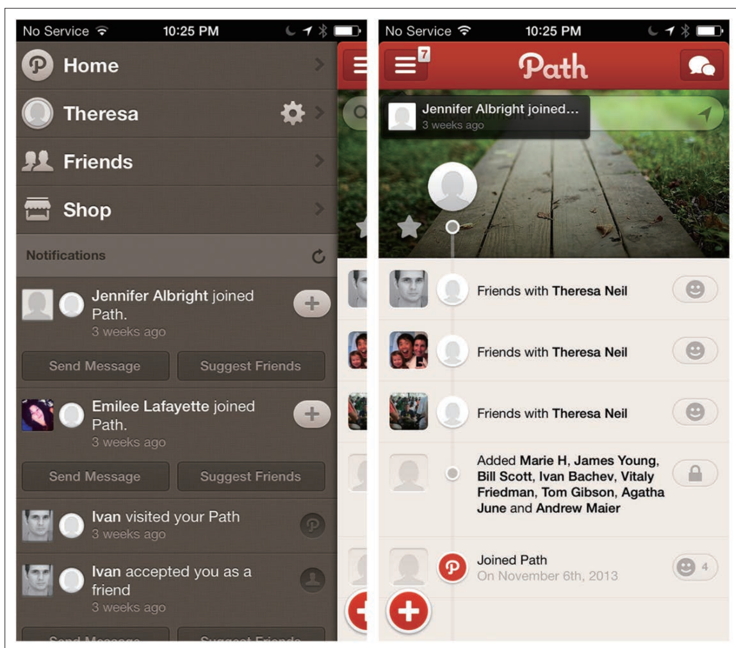



图 1-43
iOS 版 Path：点击导航图标或水平移动以显示侧边抽屉嵌入层，嵌入层将主页推向另一侧

让用户知道侧边抽屉存在的最好办法是什么？

安卓设计指南（http://developer.android.com/design/patterns/navigation-drawer.html）中建议，在用户首次打开应用时显示抽屉打开的状态，这样用户就可以看到菜单，并知道如何关闭抽屉：

> 在首次启动应用时，通过自动打开抽屉的方法向用户介绍导航抽屉。这样做可以确保用户知道导航抽屉的存在，并促使他们探索抽屉里的内容来了解你的应用结构。随后几次启动应用时，继续显示抽屉，直到用户主动地手动打开导航抽屉为止。一旦知道用户已经学会如何打开抽屉，你就可以将启动应用时的导航抽屉设置为关闭状态。

听上去不错，对吧？然而，根据我对多位客户所做的用户测试显示，这项建议并没有获得良好的效果。于是我推荐了一个类似菜谱大全（Allthecooks）那样的设计——只在首次启动应用时弹出抽屉。

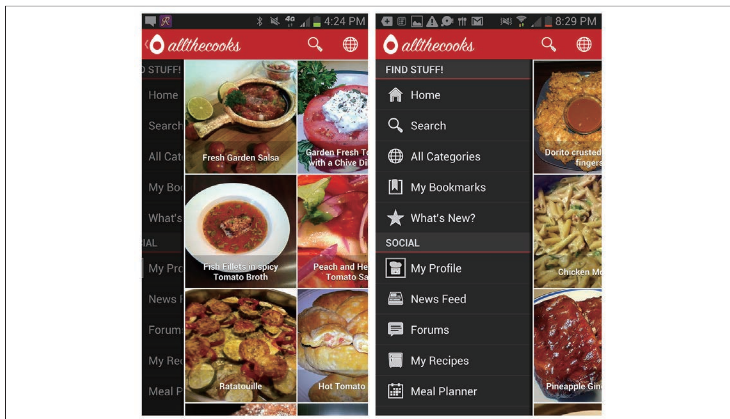



图 1-44
安卓版 Allthecooks：首次启动应用时弹出抽屉，提醒用户那里有导航菜单（http://youtu.be/iLlCs-gyNaE）

最流行的设计趋势是左侧抽屉，但也可以设计在右侧，就像 IfThis ThenThat（IFTTT）那样，又或是两侧都有抽屉，比如 Windows Phone 版的 Facebook Beta。

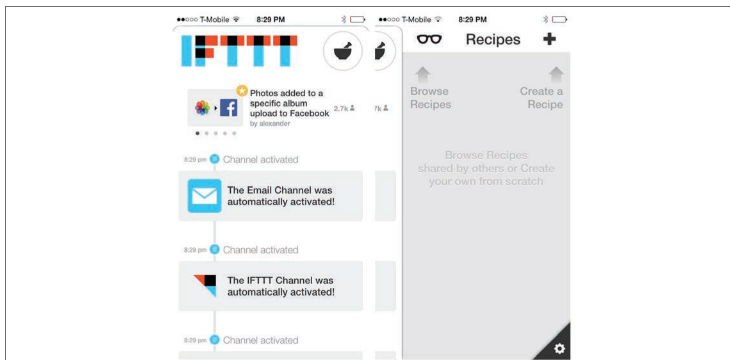



图 1-45
iOS 版 IFTTT：用于配置和导航的抽屉在右侧

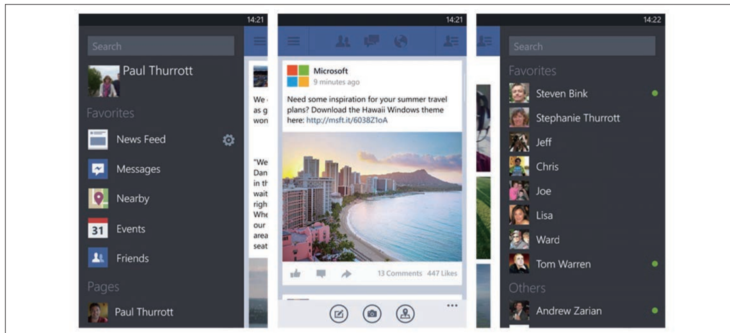



图 1-46
Windows Phone 版 Facebook Beta：两侧抽屉——左侧是主菜单，右侧是快捷链接

可不要像 om finder 和 Frost 应用那样把抽屉放在底端。这个位置与唤出 iOS 7 控制中心时使用的向上滑动手势相冲突。

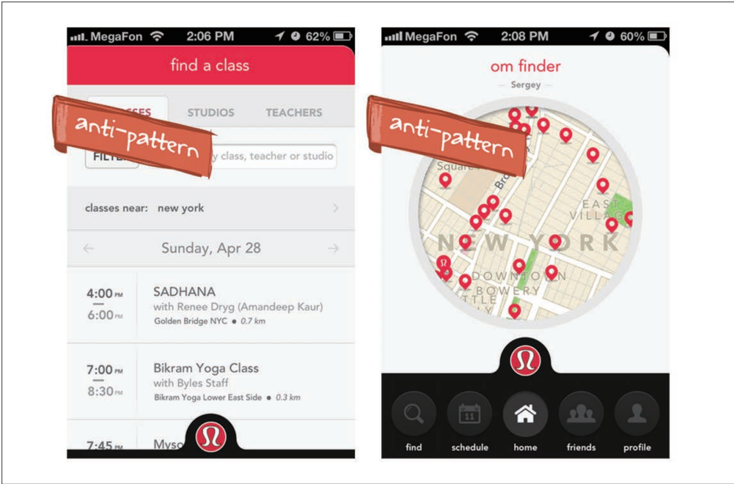



图 1-47
iOS 版 om finder：屏幕底端的抽屉与 iOS 7 的控制中心冲突

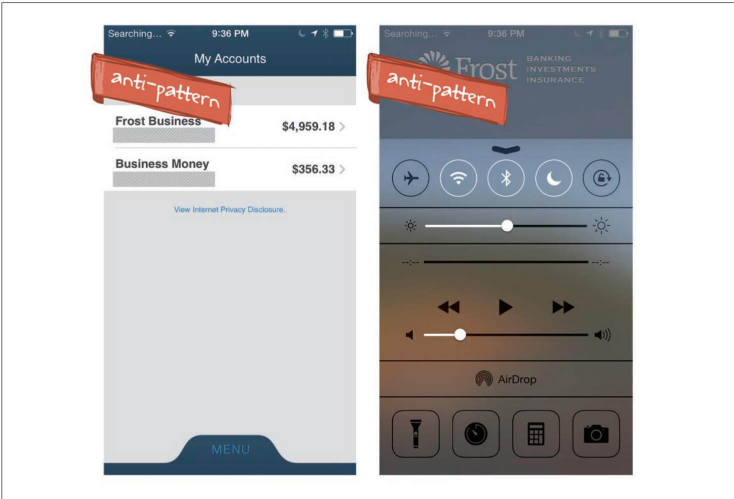



图 1-48
iOS 版 Frost：几乎每次我想打开菜单的时候，打开的都是 iOS 的控制中心

侧边抽屉的内容不必限制在导航选项范围内。Zillow 房贷市场（Zillow's Mortgage Marketplace）应用中的抽屉显示了实时房屋抵押贷款利率曲线图，像 LinkedIn 这样的社交应用也时常在抽屉中放入个人资料信息。

图 1-49
iOS 版 Zillow 房贷市场：侧边抽屉同时包含内容和菜单项

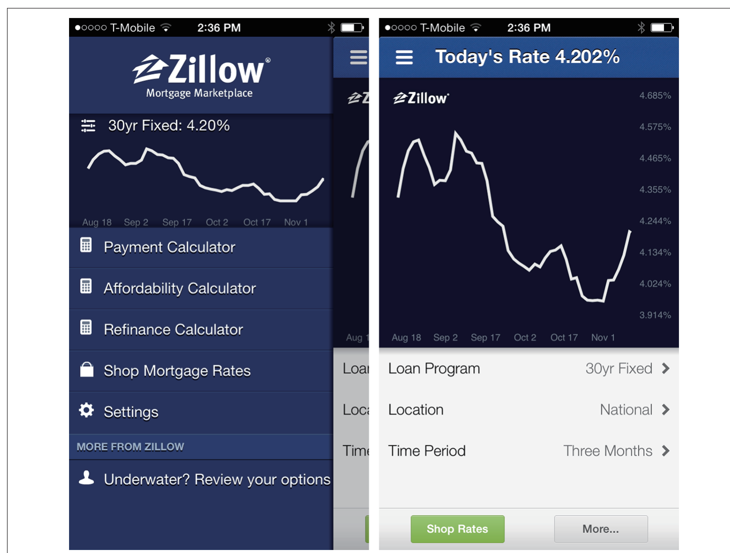
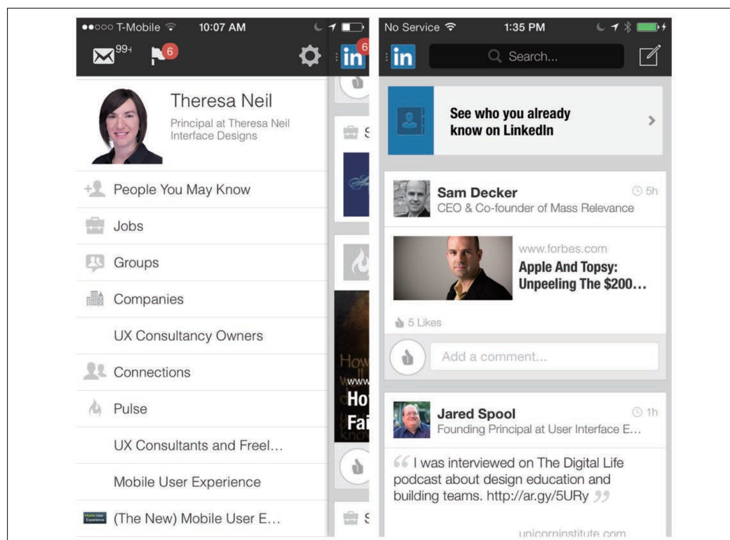



图 1-50
iOS 版 LinkedIn：侧边抽屉包含个人资料信息（哦！我有好多邮件了）




侧边抽屉式可以拥有多个层级。比如，在 Fancy 中你可以一路点击、点击再点击，直至到达最底层的类别。当你逐级深入的时候，右侧内容也会随之更新。在 Wish 的侧边抽屉中，类别路径只有两级，选择类别的模式采用的是跳板式风格。

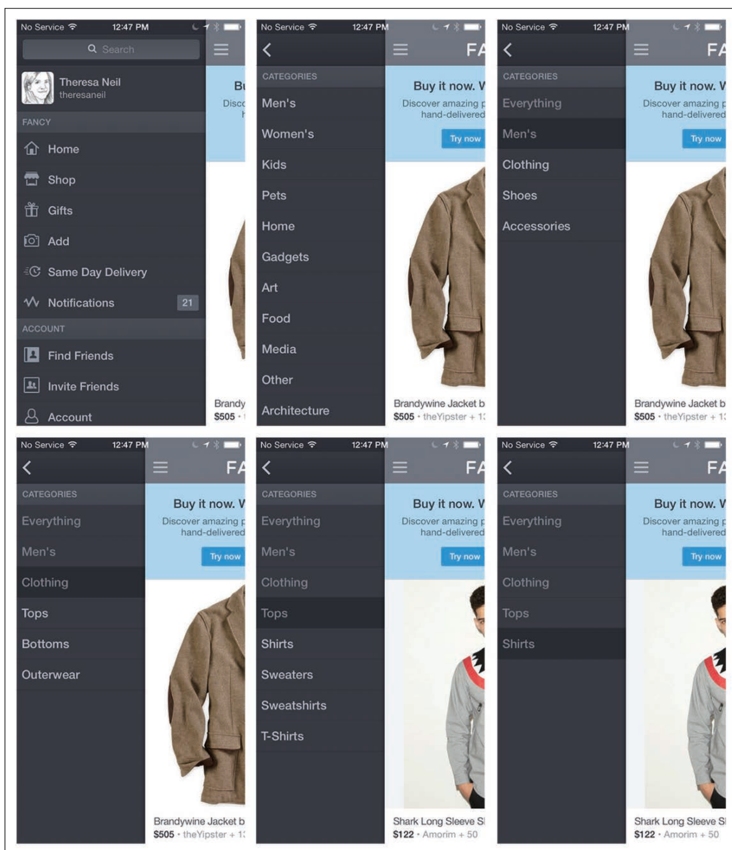



图 1-51
iOS 版 Fancy：多层级的侧边抽屉

侧边抽屉还可以让用户在高层级的场景中进行切换。在 iOS 版 Gmail 的抽屉里，点击我的名字旁边的箭头会在菜单选项下滑动打开一个面板。在这里我可以切换电子邮件账户，或是添加一个新的账户。

［小贴士］

侧边抽屉的设计可以多样化，但切忌不要让它承载太多功能。抽屉的首要功能是展示主导航选项。

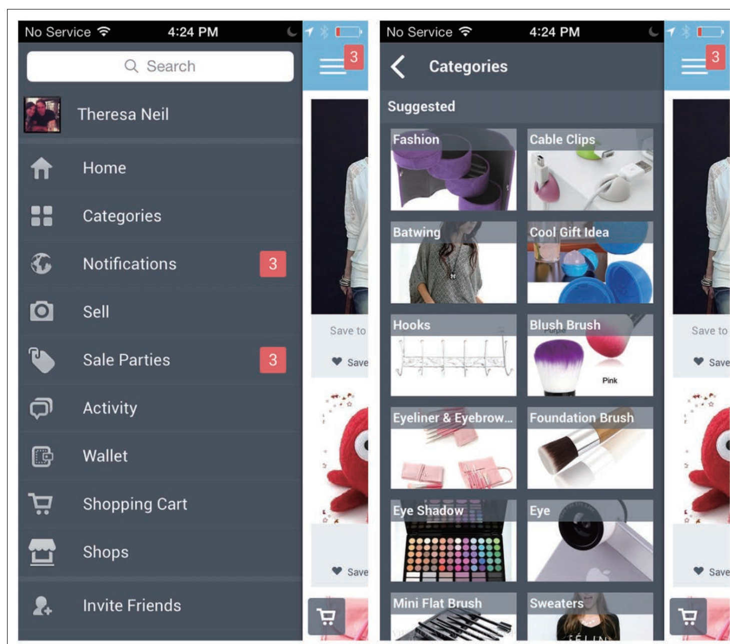



图 1-52

iOS 版 Wish：侧边抽屉的类别路径只有两级，随后转换到跳板式

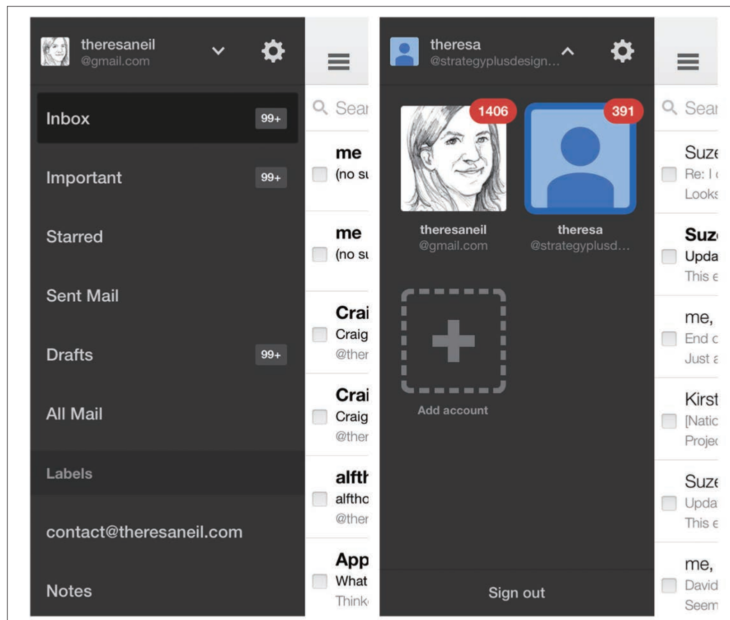



图 1-53

iOS 版 Gmail：抽屉里的下拉面板可以切换或添加账户

## 新兴模式

为了响应 iOS 7 设计规范，Luvocracy 的设计师一直在试验一种嵌入式侧边抽屉的变种（http://uxmag.com/articles/adapting-ui-to-ios-7-the-side-menu）。

点击导航图标（或者叫“汉堡包”图标）打开侧边抽屉的时候，嵌入式抽屉不仅把上一级页面向右推开，还采用 3D 效果将其推到后面。

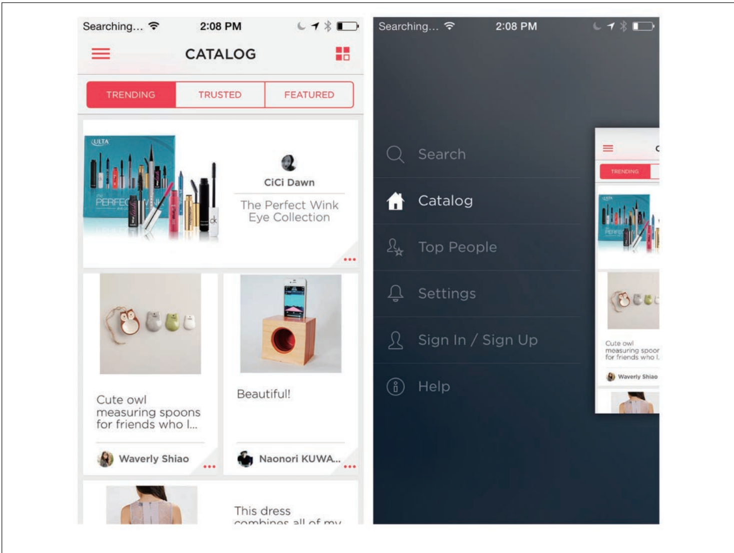



图 1-54
iOS 版 Luvocracy：点击导航或“汉堡包”，侧边抽屉会将上一级页面推到旁边和后面

Luvocracy 的导航简洁，转场流畅。然而，iOS 版 Airbnb 虽然具有类似风格的菜单，却成了一个反模式。这个新的设计拥有 3 个高层级菜单类别：旅行（Travel）、业主（Host）和登录（Log In）。“旅行”的标题横穿屏幕最上方（实际的旅行菜单选项与之分离，显示在下方），而“业主”和“登录”则放在屏幕底端，一左一右。点击“业主”或者垂直滑动会将“业主”切换到上方，而“旅行”移到底端（登录保持不动）。再次滑动会将其切换回去。这个设计对用户来说既不切实际又效率低下。我并不是完全同意 iOS 设计指南（https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/Alerts.html）的每一条规范，但是将下文中的“滚动”换成“滑动”，就指明了这个案例的问题所在：

不要让用户通过滚动来查看所有菜单选项。这会使用户不安，因为他们必须花更多的时间去分辨这些选项。而且，在滚动屏幕的时候，难免会无意识地点击选项。

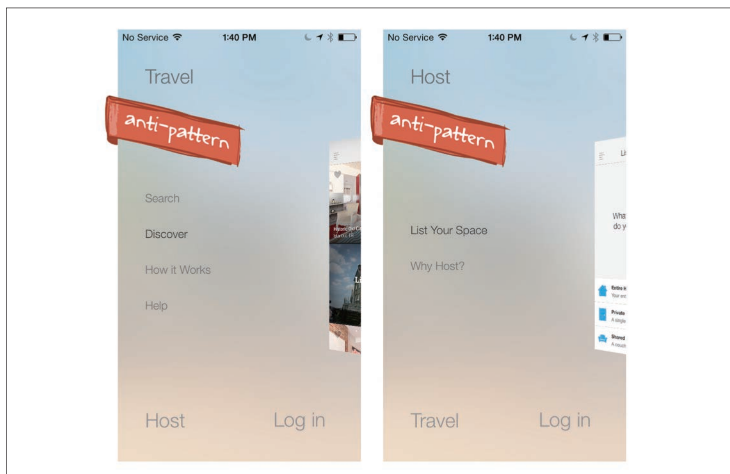



图 1-55

iOS 7 版 Airbnb：新型侧边抽屉的运用令人迷惑（http://www.youtube.com/watch?v=Rl1ZwXINhIc）

[小贴士]

在采用侧边抽屉做导航前，先制定出应用的信息架构，并与用户确认。如果侧边抽屉行得通，你应该做出几种不同的应用原型进行测试，看看哪一个设计效果最好。

安卓版本的 Airbnb 同样采用了抽屉导航，相比之下，这个版本的导航就清晰多了，无需用户费心揣测。

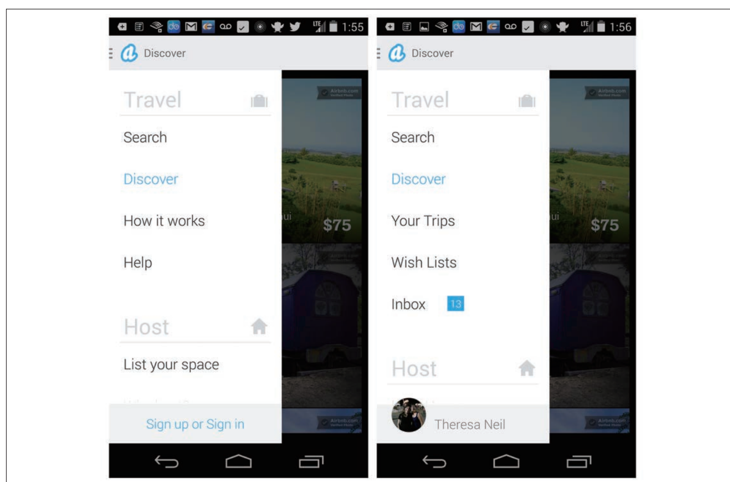



图 1-56

安卓版 Airbnb：侧边抽屉导航十分清晰

侧边抽屉导航在 iOS 上实现的一个更好的例子是 American Airlines，通过侧边抽屉导航，它展示了一个设计优美的分组菜单。

但我希望它的上级页面不是半透明的，因为那个页面仍是一个触摸对象。

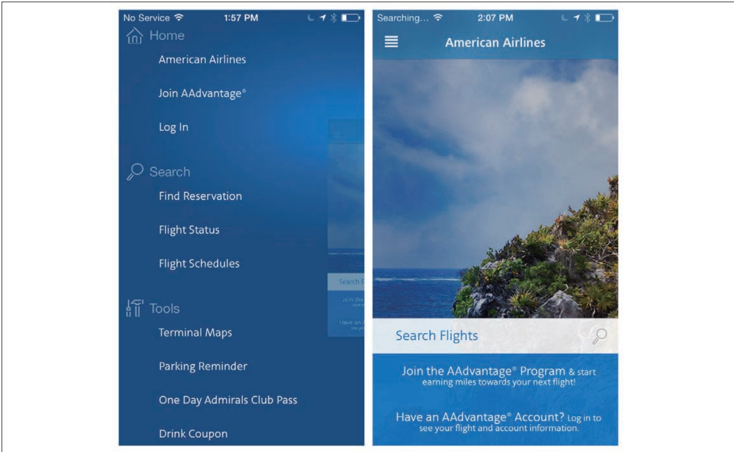



图 1-57

iOS 版 American Airlines：侧边抽屉的分组菜单设计优美，但上级页面应该是不透明的

## 下拉菜单式

在本书第 1 版中，我沿用了网页设计的叫法，把这种模式称为超级菜单（Mega Menu）。从那以后，移动网页和响应式网页设计推动了这一模式的发展，现在更普遍的叫法是下拉菜单。

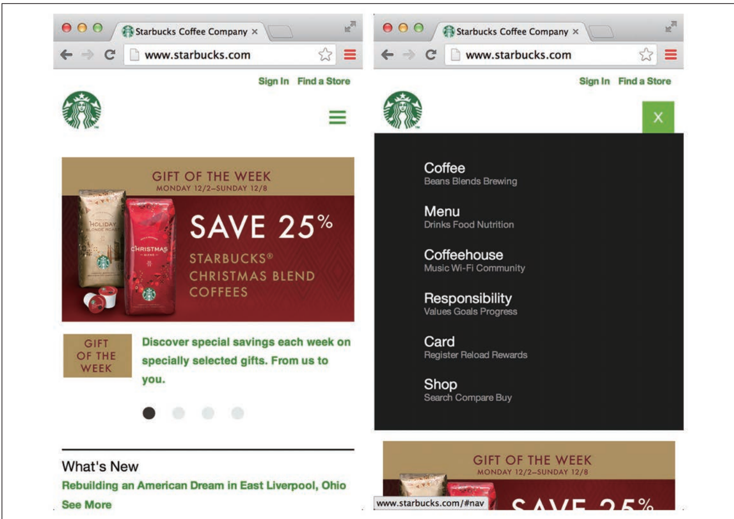



图 1-58

Starbucks.com：采用下拉菜单式导航的响应式网页设计

与侧边抽屉式一样，下拉菜单也可以是嵌入式的，将页面内容推到菜单之下；还可以是浮层式的，菜单层浮在内容页之上，如 Pocket[1] 和 Qwiki，Walmart 和 Home Depot。浮层设计在原生移动应用中更为常见。在 Ultravisual 应用里，浮层式下拉菜单从底端显示出来。

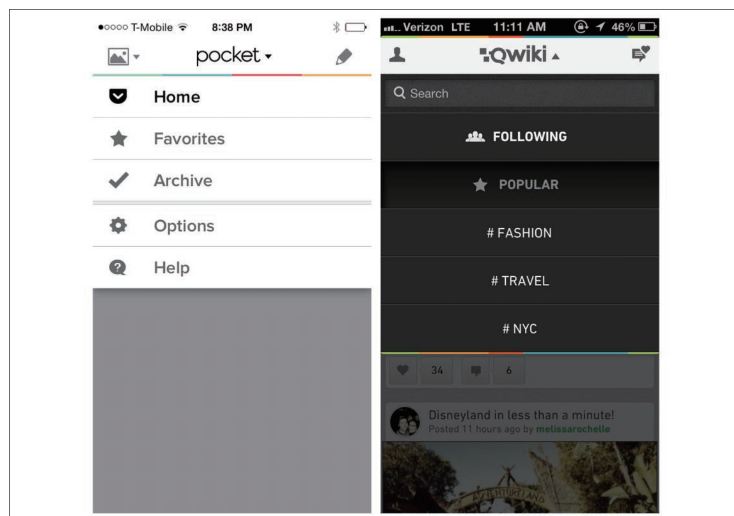



图 1-59

iOS 版 Pocket 和 Qwiki：浮层式下拉菜单

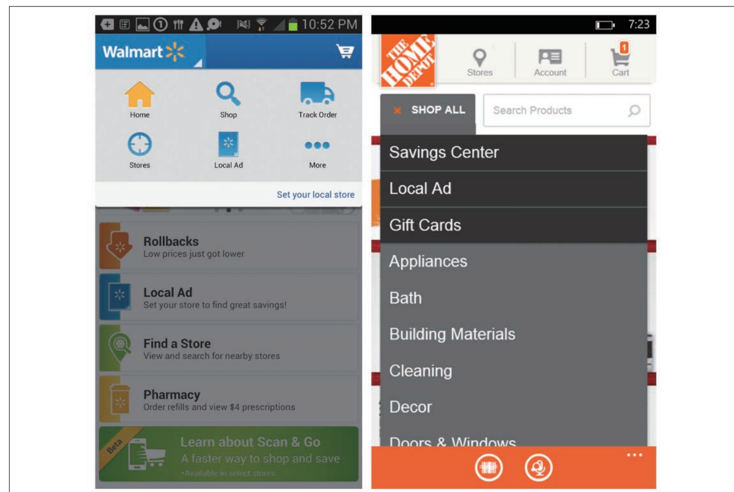



图 1-60

安卓版 Walmart 和 Windows Phone 版 Home Depot：浮层式下拉菜单

注 1：Pocket 现在的版本已经是抽屉式导航了。（译者注）

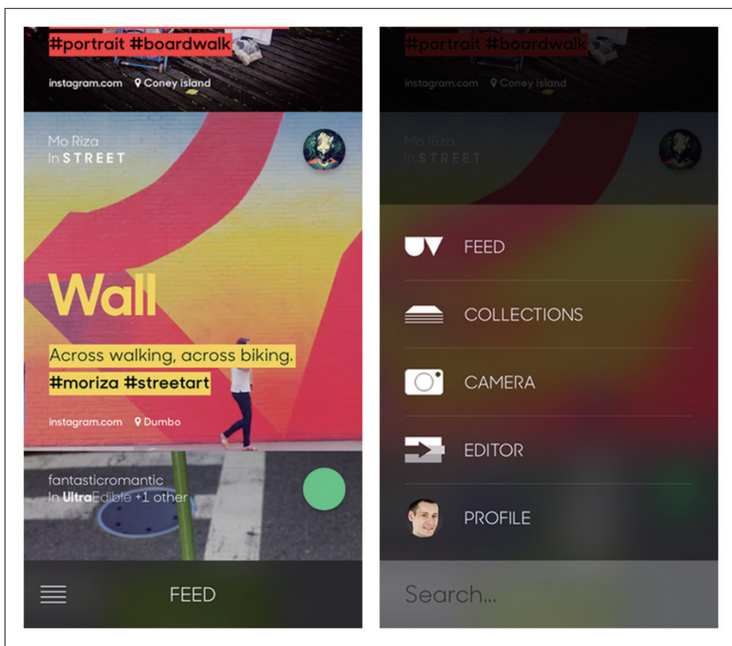



图 1-61
iOS 版 Ultravisual：浮层式下拉菜单从底端显示出来

下拉菜单式的一个重要规则是，无论什么样的手势，都能打开菜单，比如点击图标、轻滑、平移。同样，隐藏菜单也是这样的。

不要让菜单覆盖整个屏幕，要把背景露出来一点。点击背景的任一位置时，同样可以隐藏菜单。

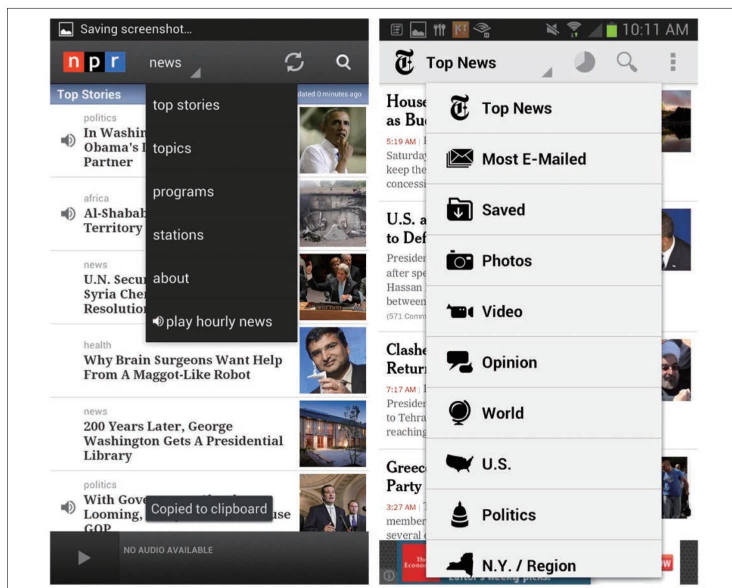
安卓为这种主导航提供了一种特别的控件——Spinner 控件。但要记住，Spinner 应该用于在同一个类别下的不同视图间进行导航，而不是跳转到完全不同的类别。

举例来说，NPR 和 NYTimes 这两个都是新闻类应用，Spinner 提供了不同的方式对应用中的海量新闻内容进行细分。但是，如果 NPR 要在这个菜单里提供其他选项，比如音乐或天气，那么就应该按照安卓设计指南中讲的，采用标签栏或是导航抽屉。

在安卓平台上，如果要通过下拉菜单实现同一类别下不同视图间的导航，就要使用 Spinner 控件。但请记住，在 iOS 和 Windows Phone 平台上，下拉菜单是自定义控件，这可能需要更多的时间去实现、测试和维护。

图 1-62

安卓版 NPR 和 NYTimes：Spinner 下拉菜单用来切换同一类别下的不同视图

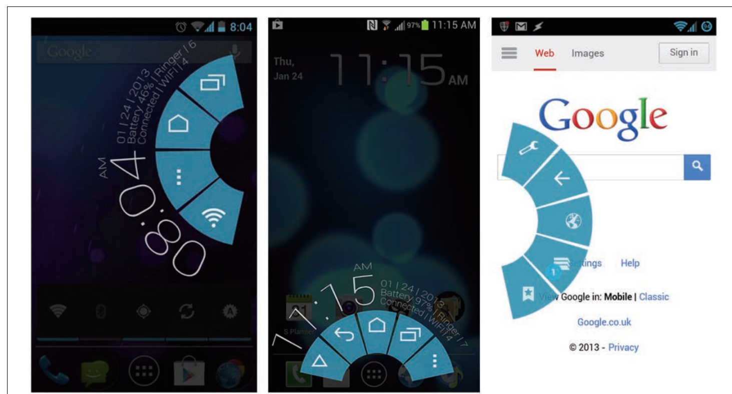



## 转盘菜单式

转盘菜单也叫轮盘、环形菜单或径向菜单，自 20 世纪 90 年代以来一直出现在桌面软件设计中，近年来更多地出现在网页应用中。在游戏设计中，这种模式也非常受欢迎。所以，当上手试玩 PIE（开源的第三方安卓系统 Paranoid 中的转盘菜单）时，我非常兴奋，并且对其操作之流畅感到非常惊讶。

图 1-63

开源的第三方安卓系统 Paranoid 中的转盘菜单式导航（http://www.androidpolice.com/2013/01/22/paranoid-androids-pie-navigation-system-is-almost-out-of-the-oven-and-it-smells-fantastic/）



如果安卓系统把转盘菜单设为标准控件之一的话，那么我不认为它适合用来作应用的主导航，因为这会在操作系统和应用之间造成太多的冲突。

看看其他操作系统吧，我发现只有屈指可数的几个应用尝试采用了这种模式作为主导航。我找到的这些例子并不那么理想。比如说，使用 PortalWebBrowser 的多层轮盘导航时，用户需要具备强大的心理能力以及外科医生般精准的操作手法。

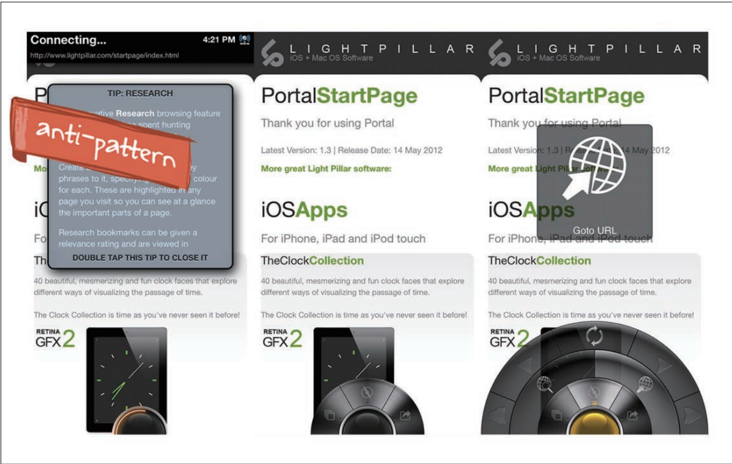



图 1-64

iOS 版 PortalWebBrowser：使用多层转盘菜单时需要外科医生般精确的触摸

［小贴士］

这个模式可能是所有主导航模式中最不常用的，在设计任何有多个层级的菜单时应该避免采用。如果你的应用信息结构扁平化，那么可以考虑选用选项卡式菜单，这对所有用户来说都更为熟悉。

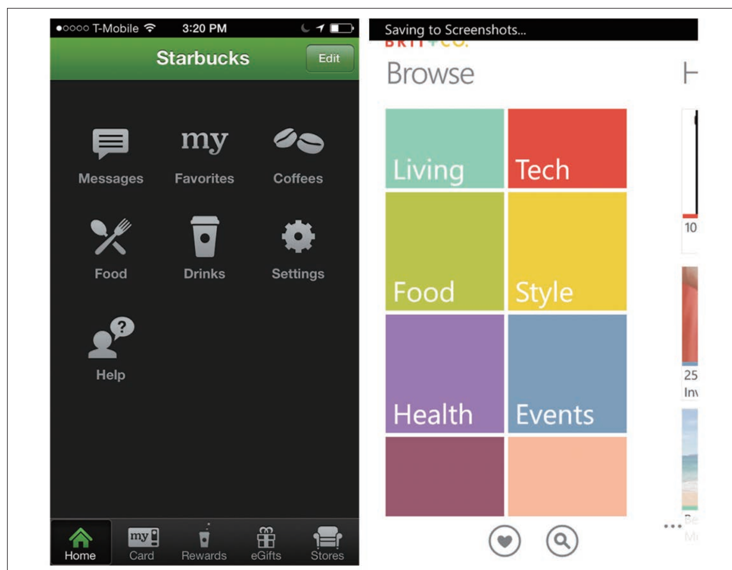
Paranoid 安卓系统中的 PIE 与 PortalWebBrowser 中的转盘菜单在设计上的主要区别在于，后者的层级太多，而且触摸对象太袖珍了。点击、长按，然后滑过轮盘，移至或大或小的楔形按钮，这个手势既不简单也不自然。不过，在选择操作方面，转盘菜单式是有一些优秀示例的，请参看第 5 章。

## 次级导航模式

如果只讲主导航的话，本章内容会不够完整，所以我将次级导航也放到了本章。我所说的次级导航，指的是在一个选中模块内活动的导航。比如说，星巴克采用选项卡菜单式作为主导航，而首页上使用的是跳板式次级导航。与之类似，Brit & Co. 也使用选项卡菜单式作主导航（即一个 Windows 透视控件），在浏览模块内则采用跳板式作为次级导航。

图 1-65

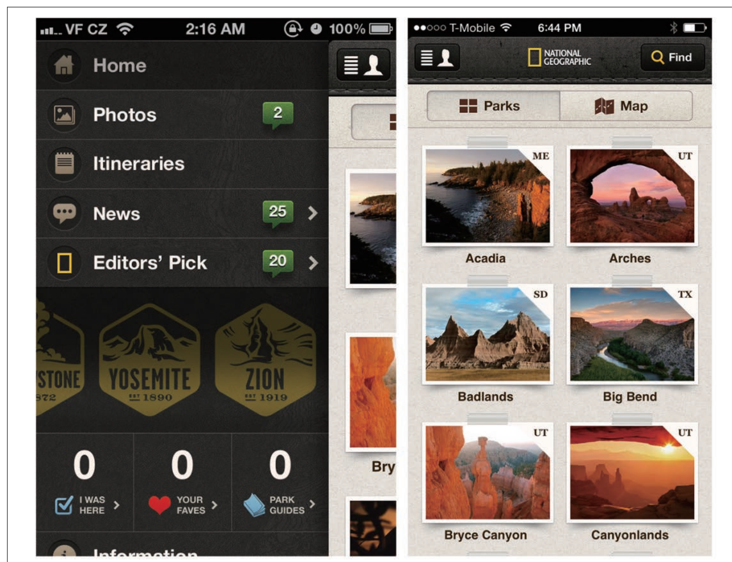
iOS 版星巴克和 Windows Phone 版 Brit & Co.：跳板式的次级导航




所有主导航模式都同样可以用作次级导航模式。选项卡式导航里包含选项卡、列表或仪表盘，跳板式导航里包含陈列馆式等，都十分常见。

图 1-66

iOS 版《国家地理》(*National Geographic*) 出品的国家公园 (National Parks)：主导航是侧边抽屉式，次级导航是陈列馆式



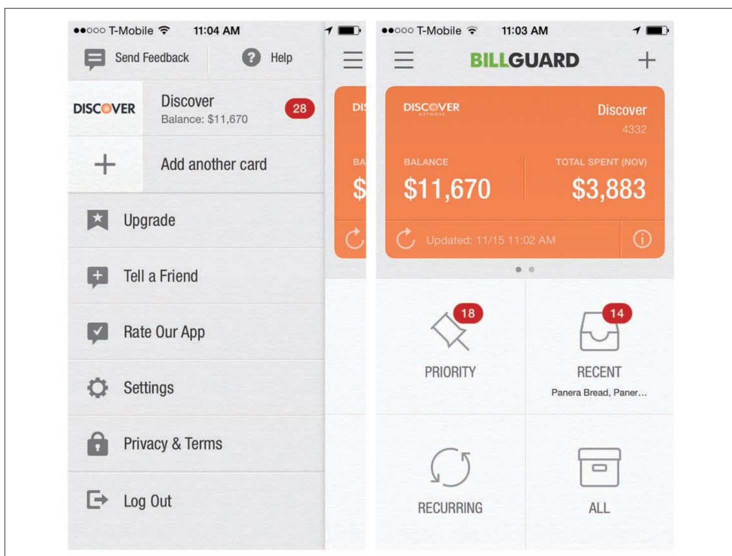



图 1-67
iOS 版 BillGuard：主导航是侧边抽屉式，次级导航是跳板式

还有一些模式也可以用作次级导航，其中有翻页式、滚动选项卡式和伸缩面板式。

## 翻页式

这种模式可以通过滑动手势快速导航内容页。表现这种导航模式的最常见方式是利用页面指示器（Page Indicator，iOS 里对一行小圆点的叫法）。卡片隐喻也同样适用于页面，比如 Ness 和 Foodspotting 中的设计。在这些例子中，部分可见的背景卡片或页面用以引导用户去滑动屏幕。

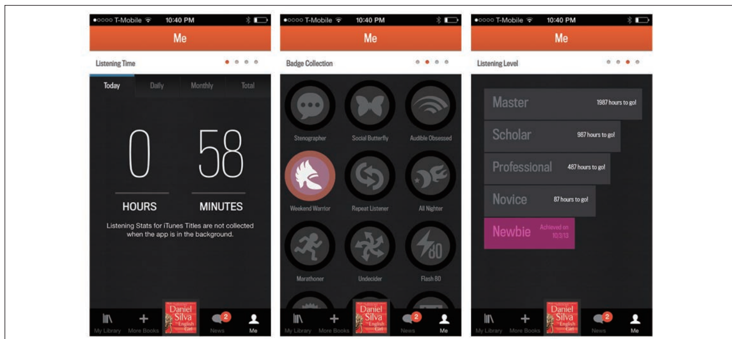
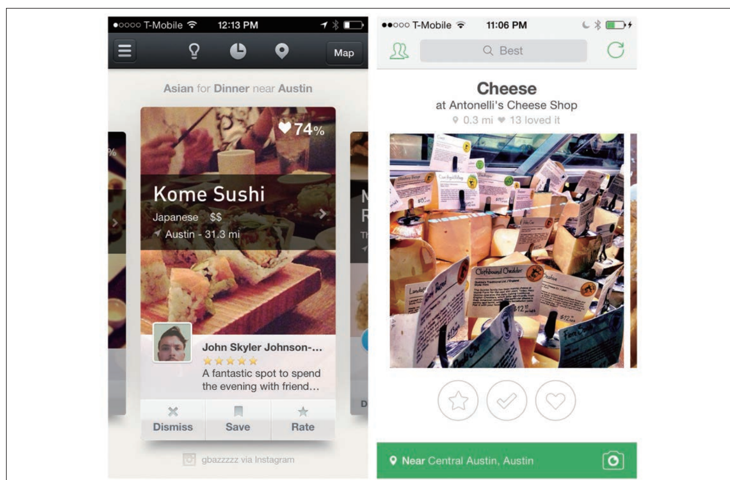



图 1-68
iOS 版 Audible：主导航是选项卡式，次导航是翻页式。注意页面指示器

图 1-69

iOS 版 Ness 和 Foodspotting：部分可见的内容引导用户滑动屏幕

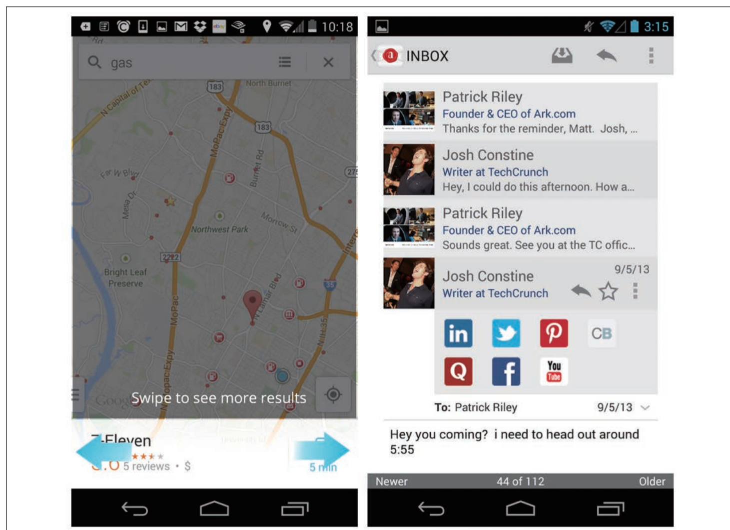



许多安卓应用都提供了类似的翻页体验。在谷歌地图（Google Maps）中，你可以滑动搜索结果列表。初次使用时，会显示一个关于这种滑动翻页操作的提示。

Ark Mail 与 Gmail 类似，都用翻页式作为电子邮件消息的快速导航。一窄行页脚显示消息的总数和当前消息数，并在底端两侧分别标示出新旧邮件，以此表明水平滑动可以看到其他邮件。

图 1-70

安卓版谷歌地图和 Ark Mail



News360 是翻页式的一个反模式案例，因为它缺乏功能指示。如果它像 HuffPost 那样，在顶部加入页面指示器小圆点标识而不是采用垂直滚动条，便会立刻让人明白，使用水平滑动手势便可以查看下一篇新闻。

[小贴士]

运用翻页式时要利用移动的手势控制，不要依赖像下一步按钮或标签页这类桌面端模式。但要注意提供视觉指示，让用户知道可以使用滑动手势。

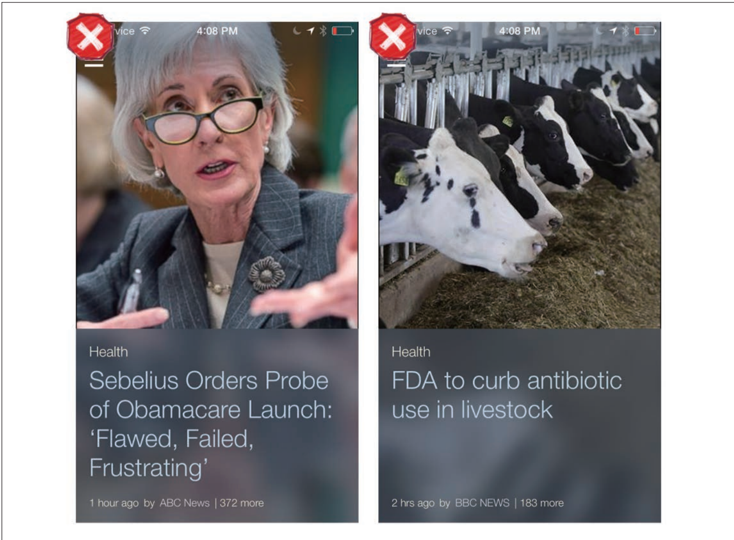



图 1-71
iOS 版 News360：水平滑动查看下一篇文章的提示信息不可见

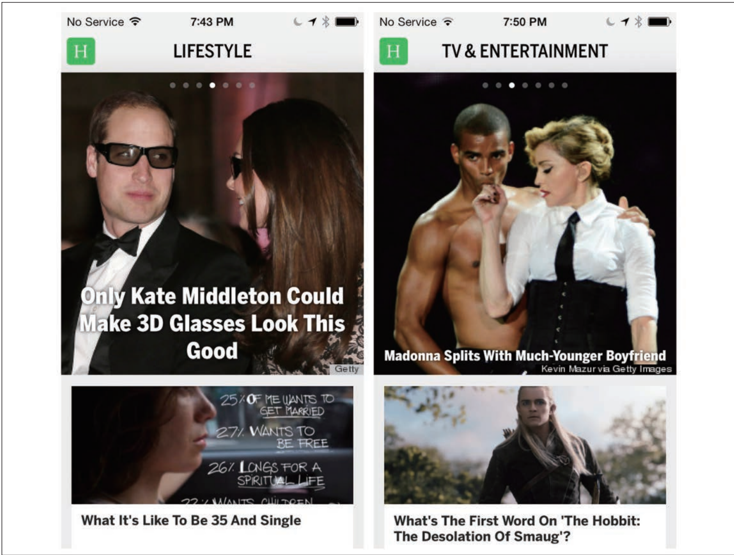



图 1-72
iOS 版 HuffPost：位于顶部的小巧的翻页指示器标识出滑动功能

## 滚动选项卡式

安卓设计指南（http://developer.android.com/design/building-blocks/tabs.html）把这些次级导航控件称为滚动选项卡式，所以我延续它的叫法。在某一个模块中显示多种类别或视图时，这种模式非常有用。滚动选项卡一般比标准的标签栏要窄，因为它们没有必要作为触摸对象。更典型的是将它们用作水平滑动指示。

在 Google Play 应用中，在导航抽屉中选择了一个菜单项（比如电影和电视剧）之后，滚动选项卡用来对结果进行筛选。Songza 和 TuneIn 的例子展示了这种模式的其他应用。

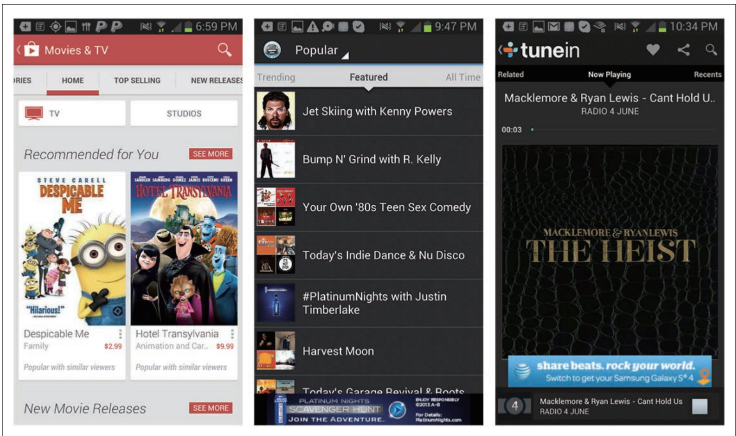



图 1-73

安卓版 Google Play、Songza 和 TuneIn：滚动选项卡式的次级导航

如果你要使用这种模式，请确保你的界面设计清晰地指示出了选中的标签项。

## 折叠菜单式

折叠菜单式（Accordion，又称手风琴式）能使用户在同一页面上查看到更多的信息。这种模式比跳转到另一页面再跳回去更高效。注意 Elevatr、Flava 和安卓 Play Store 这些例子，它们全都使用了常见的图标去指示面板的展开或收起状态。

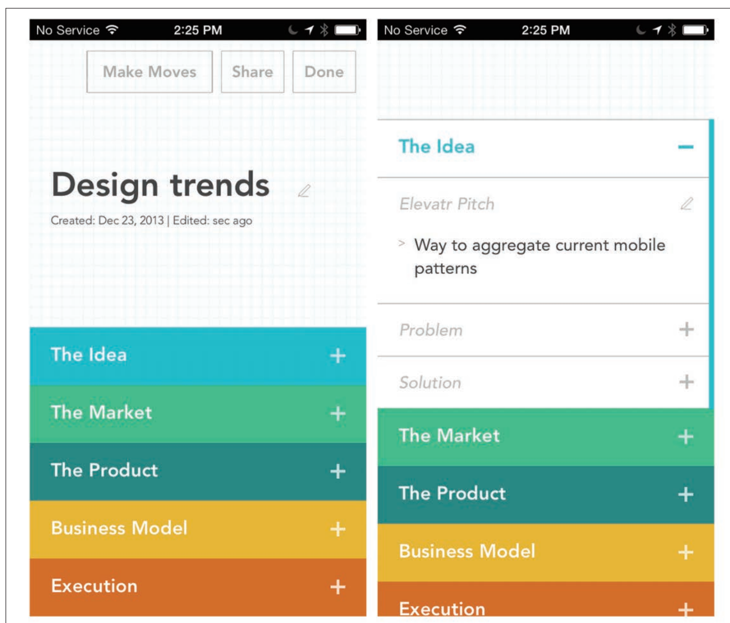



图 1-74

iOS 版 Elevatr：当面板展开时“+”号变成“－”号，而且标题变成反色

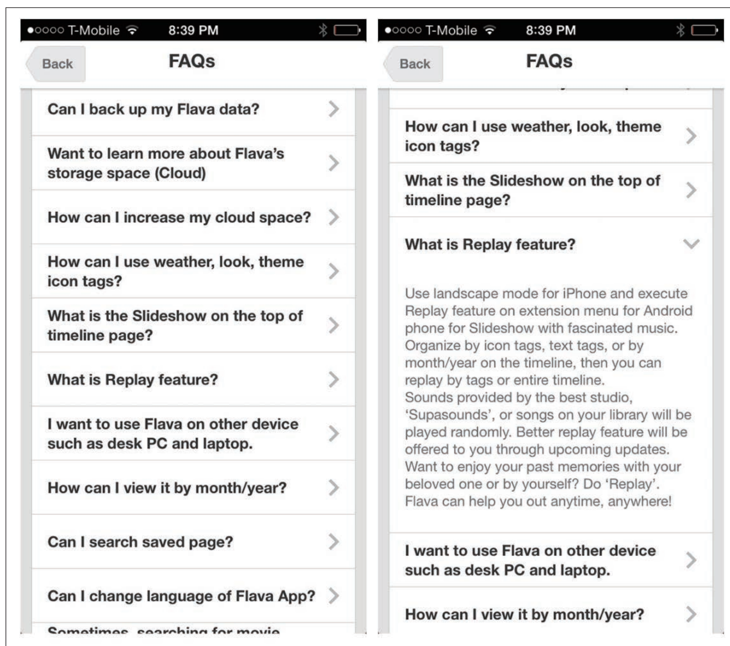
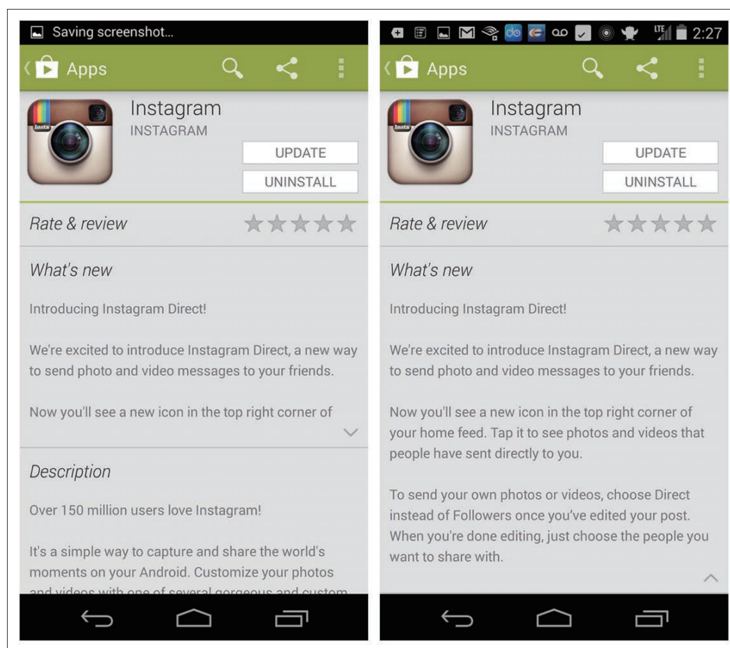



图 1-75

iOS 版 Flava：向下的箭头指示面板展开

图 1-76

安卓 Play Store：“新版变化”面板中的展开 / 收起




[小贴士]

要使用常见的图标来指示折叠式菜单的打开或关闭状态。

# 第 2 章

# 表单




- 常见表单模式

  登录表单（Sign In）、注册表单（Registration）、个性化注册表单（Registration with Persona lization）、结账表单（Checkout）、计算表单（Calculator）、搜索表单（Search Form）、多步骤表单（Multi-Step）、长表单（Long Form）

多数网页应用都大量依赖表单实现数据输入和配置。尽管我们对高效的表单设计已有一些研究成果和设计策略，但网络上糟糕的表单仍然随处可见。设置在线账号、购买商品、提交申请、填写调查问卷等，简直让我们筋疲力尽。

图 2-1

Windows Phone 版 American Airlines 机票预订：差劲的表单设计

有时我们能顺利提交表单，但常常中途放弃。表单流失（指的是那些还没完成并提交表单就放弃的用户）是一个非常严重、代价高昂的问题。但与移动端相比，网页端的问题还算不上什么，因为狭小的屏幕和受限的输入方式导致糟糕的移动表单设计简直没有留存的余地。

因此，在你设计任何移动端表单之前，我强烈推荐你阅读以下资料来复习关于表单设计的基本技巧：

- 《Web 表单设计：点石成金的艺术》（*Web Form Design: Filling in the Blanks*，http://www.amazon.com/Web-Form-Design-Filling-Blanks/dp/1933820241，Luke Wroblewski 著，由 Rosenfeld Media 于 2011 年出版）
- Forms On Mobile Devices: Modern Solutions（http://www.smashingmagazine.com/2010/03/11/forms-on-mobile-devices-modern-solutions/，Luke Wroblewski 著）
- Better Mobile Form Design（http://www.lukew.com/ff/entry.asp?1014，Luke Wroblewski 著）
- Mobile Form Design Strategies（http://www.uxbooth.com/articles/mobile-form-design-strategies/，Chui Chui Tan 著）
- Mobile Inline Form Validation（http://www.uxmatters.com/mt/archives/2012/09/mobile-inline-form-validation.php，Steven

Hoober 著）

- Removing Stumbling Blocks in Mobile Forms（http://www.smashingmagazine.com/2012/05/03/removing-stumbling-blocks-in-mobile-forms/，Robert Brauber 著）

作为对上述资料的补充，下面这些表单设计模式将有助于你设计出让用户满意的应用。这些模式包括：

- 登录表单
- 注册表单
- 个性化注册表单
- 结账表单
- 计算表单
- 搜索表单
- 多步骤表单
- 长表单

## 登录表单

登录表单应该包含尽可能少的输入：用户名、密码、操作按钮、密码帮助以及注册选项。一些应用将这些输入框全部设计在同一页面上，比如 Remember the Milk 和 Groupon。

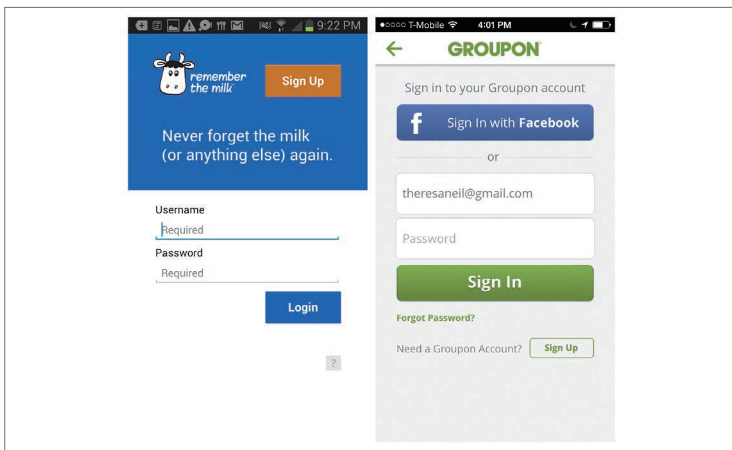



图 2-2
安卓版 Remember the Milk 和 iOS 版 Groupon：最简洁的登录框

另外一些应用，比如 Instagram，则首先显示登录和注册选项，然

后引导用户去填写相应的表单。你应该让第一个输入框自动获得焦点，避免用户做额外的点击操作以打开键盘。

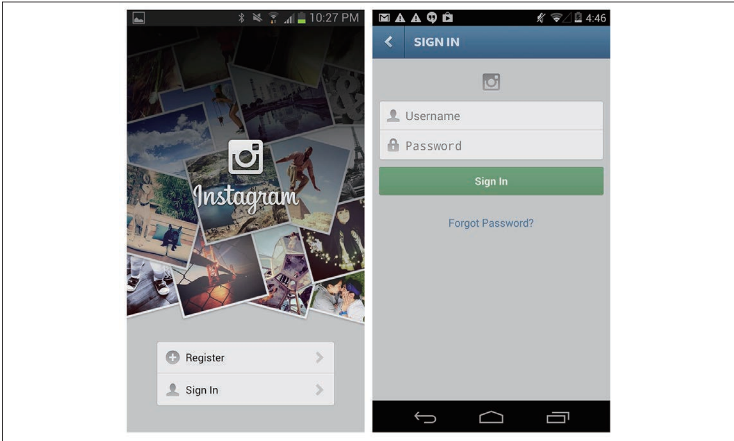



图 2-3
安卓版 Instagram：显示在最前面的注册和登录选项

登录表单本身应该利用一种新技术——明文显示已输入的密码，这项技术已获得了可用性测试与现场使用效果的有力支撑。比如 LinkedIn 的登录输入框就明文显示密码，同时也为用户提供选项来隐藏密码。开启和关闭图标是通用的，或者采用文字“隐藏 / 显示”（参见“注册表单”一节中关于 Polar 的示例）。

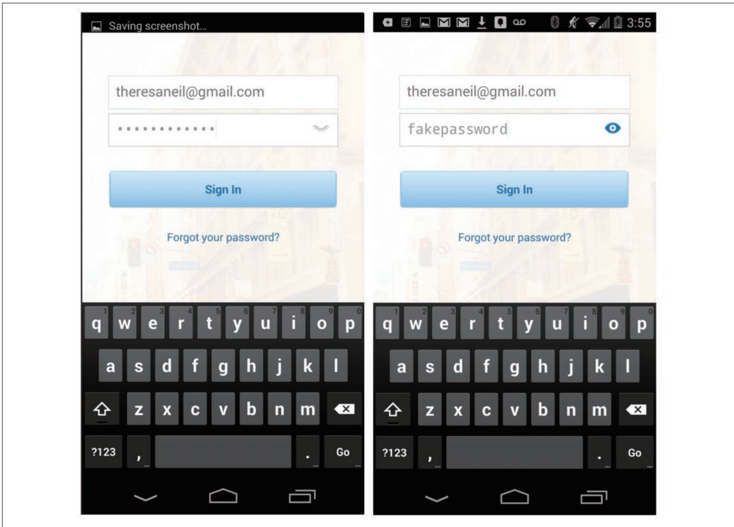



图 2-4
安卓版 LinkedIn：明文显示密码的选项

其他一些应用，比如 Wunderlist 和 Box，将注册和登录选项放在引导页上。通过在每一页上固定显示这些选项，让新用户一旦决定使用应用就可以立刻注册，同时老用户也一样可以方便地登录。

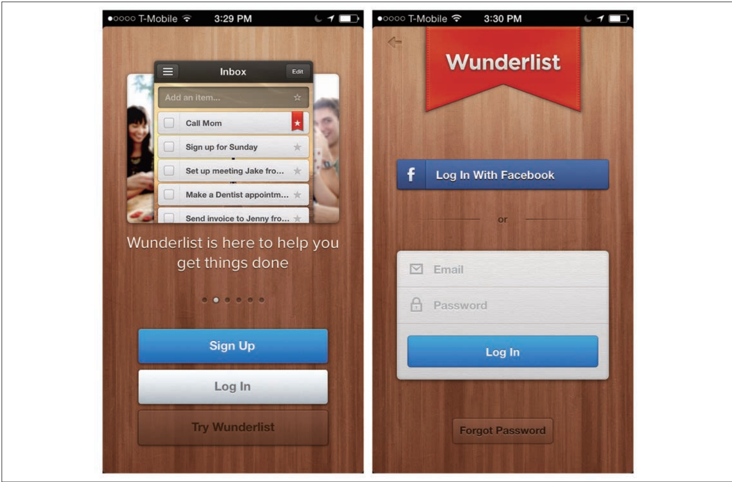



图 2-5
iOS 版 Wunderlist：注册和登录一直呈现在引导页上

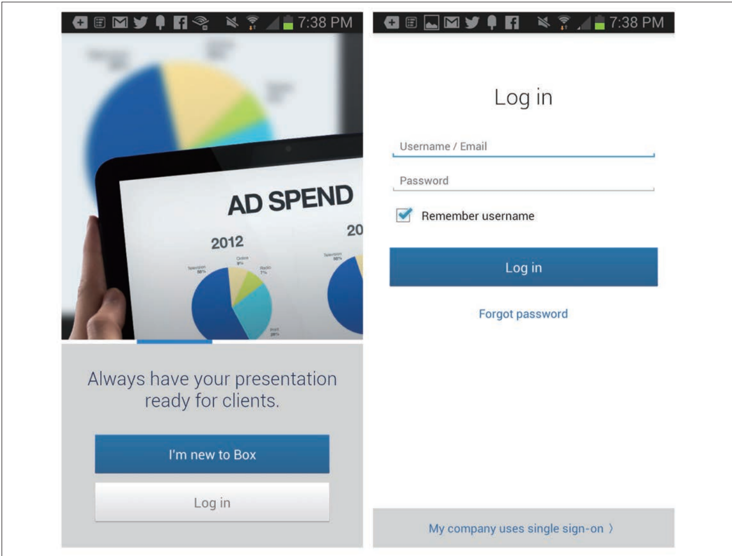
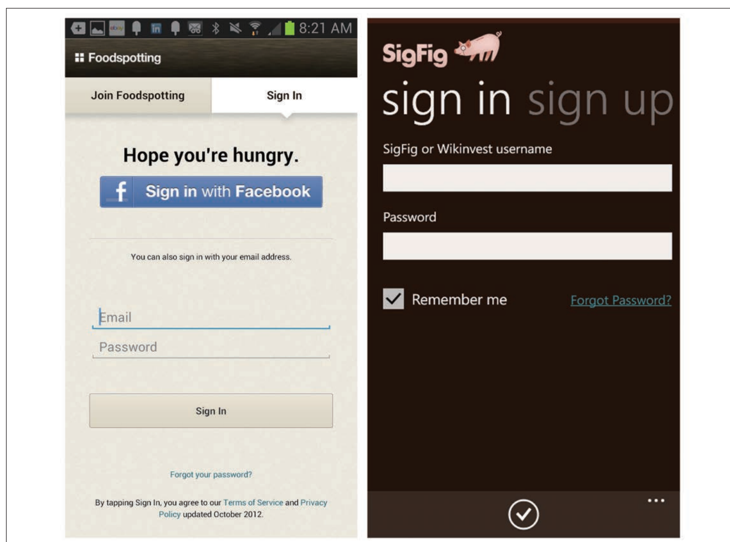



图 2-6
安卓版 Box 让用户在对引导产生兴趣时可以轻松注册

标签页是另一种既呈现两个选项又能将导航最小化的可行模式，比如 Foodspotting 和 SigFig 的设计。

图 2-7

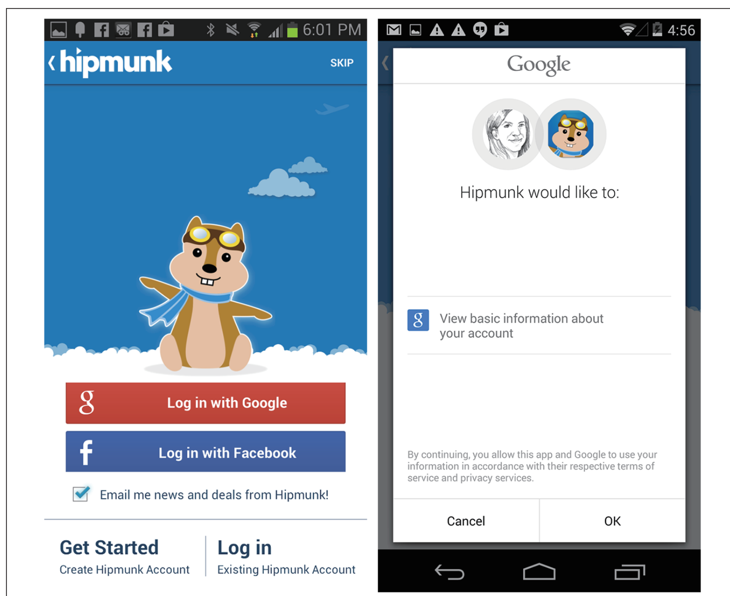
安卓版 Foodspotting 和 Windows Phone 版 SigFig：登录和注册标签页




这些模式中的任何一种都可以加上第三方登录选项，不要与第三方注册混淆，后者我们会在注册模式中讨论。第三方登录会让用户对其他的社交媒体授权，授权成功后即可进入应用。

图 2-8

安卓版 Hipmunk 提供了三个登录选项：创建账户、Facebook 登录和谷歌登录



还有一种登录模式在金融领域十分流行。这种模式直接放弃用户名输入框，只要求输入密码。在安装应用后，用户身份就可以得到验证，只要再输入密码就可以访问敏感数据。把手机的 PIN 码作为密码也能实现同样的效果，这种模式可以参考 Frost Bank 和 Personal Capital 的例子。

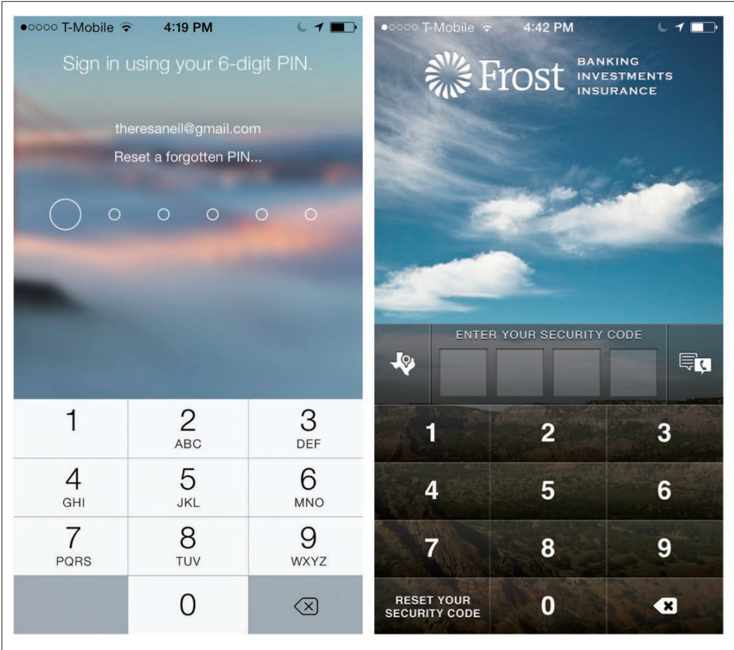



图 2-9

iOS 版 Personal Capital 和 Frost 只要求输入密码或 PIN 即可登录

到目前为止，我们看到的绝大多数例子中，登录表单在启动应用时就会显示出来。但在很多情况下，最好让用户在使用应用时无需登录或注册。

你可以畅快地浏览 Groupon 和 Etsy，直到想购买或是收藏一件商品时才需要登录（选择分享、添加到愿望清单、添加评论等时也可以要求用户登录）。为什么你应该让用户不登录或注册就能购物呢？ Jared Spool 在他的“The $300 Million Button”一文中做了解释（http://www.uie.com/articles/three_hund_million_button）。 若需要进一步的说明，请参看第 11 章中 Greg Nudelman 所撰写的“番外篇”。

**[小贴士]**

登录不一定非得是第一步。想一想什么时候验证用户身份才是最合适的。不要在登录页上进行创新，采用标准设计会更易于用户登录。

邮电

图 2-10
iOS 版 GroupOn：随意浏览，购买时再要求登录

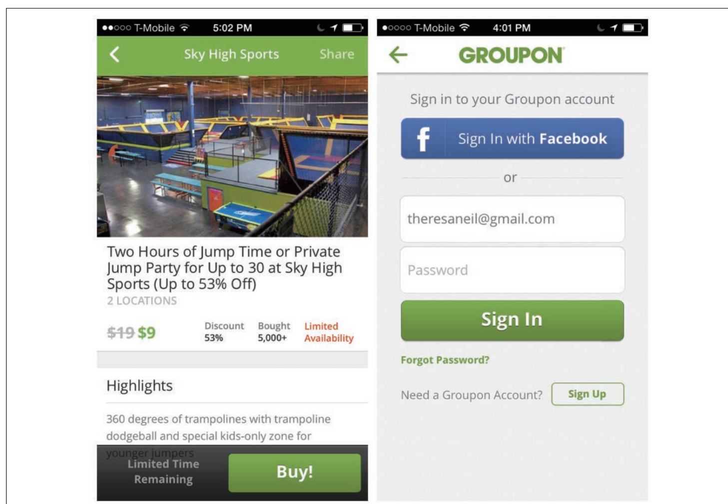
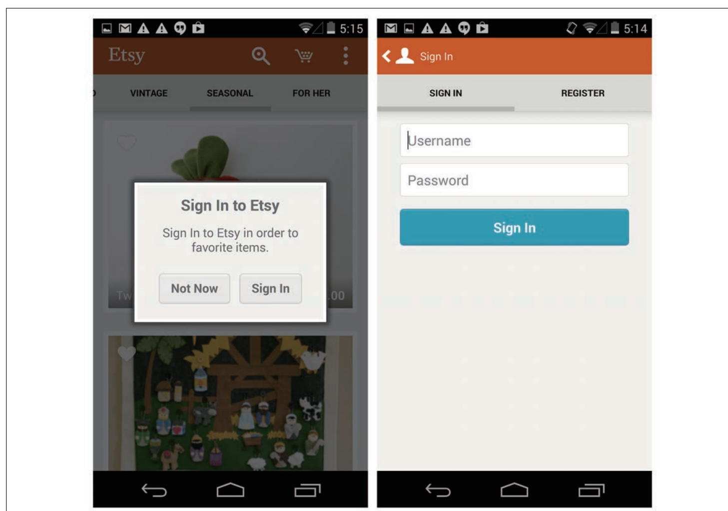



图 2-11
安卓版 Etsy：用户“喜欢”一件商品时再要求他登录




## 注册表单

企业热爱客户数据，所以他们往往试图在用户注册时获取尽可能多的信息。但注册表单应该包含尽可能少的输入项。因为每增加一项输入，就会大大降低注册表单的转化率，所以用户体验顾问兼作者 Chui Chui Tan 建议，应该毫不留情地砍掉“没有任何重要

功能的元素”。经过大量的用户测试验证，比较容易下手的是去掉不必要的确认电子邮件字段和确认密码字段，如下图的 NOOK 和 LearnVest 所示。

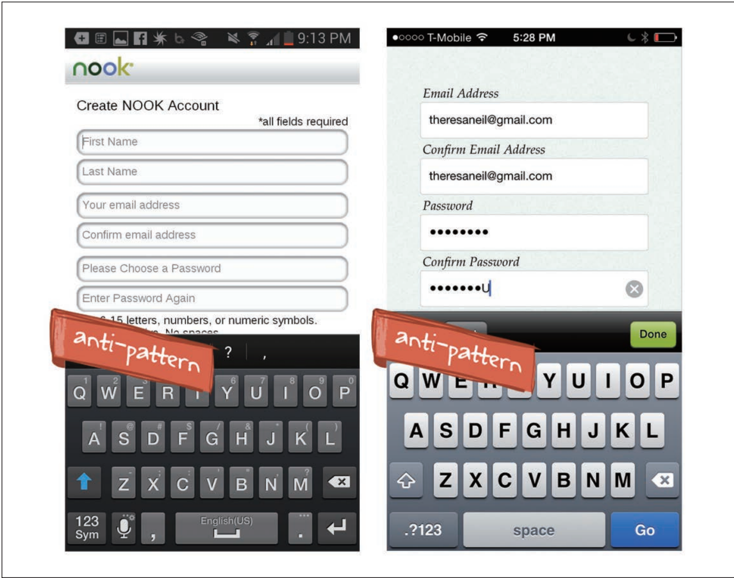



图 2-12
安卓版 NOOK 和 iOS 版 LearnVest：多余的确认电子邮件和密码的字段

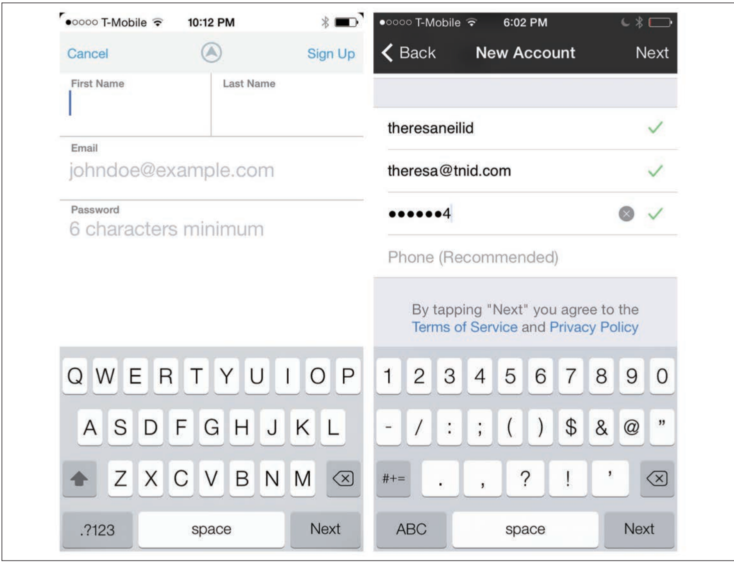



图 2-13
iOS 版 Rove 和 Kik：测试确认单行电子邮件和密码字段的可行性

确认电子邮件地址是必需的吗？ Path 的注册表单稍微长那么一点，但它却清晰地标注出了选填字段，而且用提示用户核对的办法取代了重复输入邮件地址。

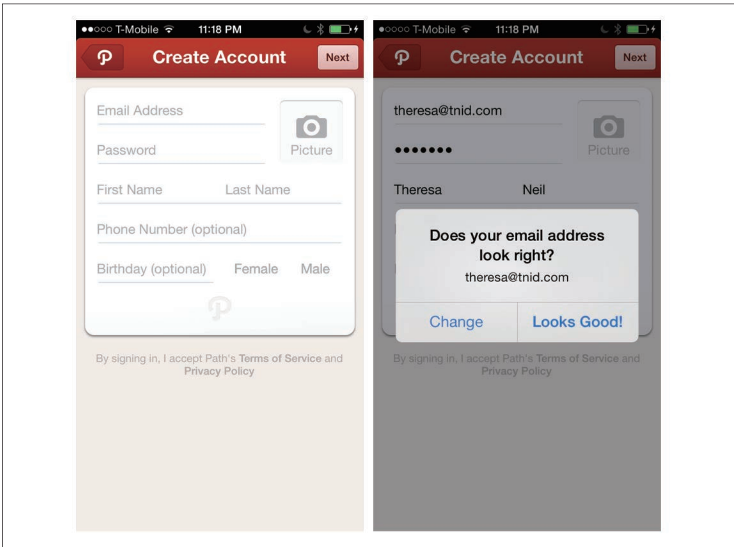



图 2-14
iOS 版 Path 清楚地标注出了选填字段，并用弹框提示用户确认电子邮件地址

注册表单可能是你为你的应用设计的第一个表单，所以你应该建立一个易于阅读的标签标准。表单大师 Luke Wroblewski 打造的有趣的投票应用 Polar 就是一个优秀的例子。注意，每个表单项都有即时反馈，而且密码项有“隐藏 / 显示”选择，这就使得重复输入密码更加没有必要。

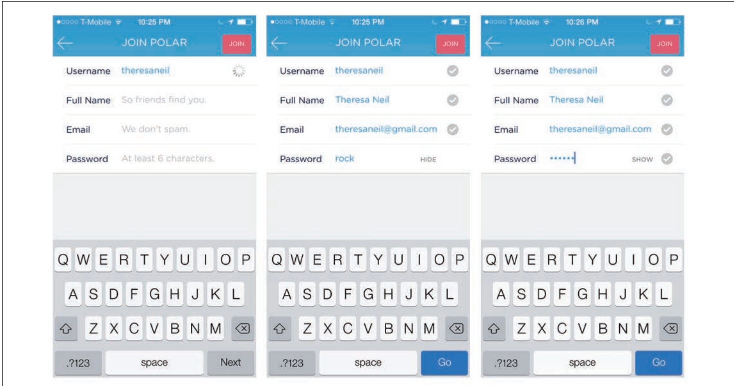



图 2-15
iOS 版 Polar 是表单布局和交互设计的优秀范例

水平对齐的表单标签可能不是最好的选择，因为这样可能会使标签显示不全，或者和表单输入框挤到一起。可以考虑使用垂直对齐的标签，比如 Fancy 和 Remember The Milk。




图 2-16
水平对齐的标签可能显示不全或部分重叠

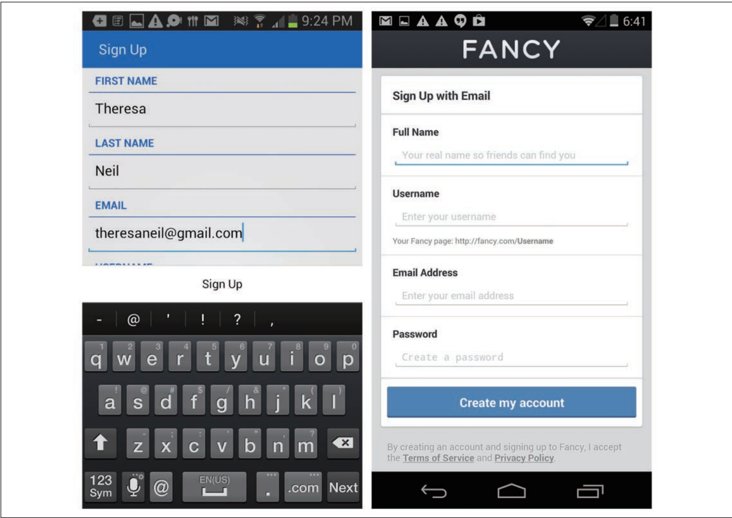



图 2-17
安卓版 Remember The Milk 和 Fancy：垂直对齐的标签往往更好一些

还有一个选择是使用提示文字（即输入框中显示有浅色的默认文字，提示用户该字段的填写要求）。但这会带来一个问题，就是一旦用户开始输入字符，提示文字就会消失。如果用户不注意的话，很容易就忘了他应该在这里输入什么。悬浮标签是一种新

型模式，可以部分解决水印式标签（即提示文字）存在的易用性问题。在 http://mattdsmith.com/float-label-pattern/ 上可查看这一模式的发明者 Matt Smith 撰写的更多相关信息。

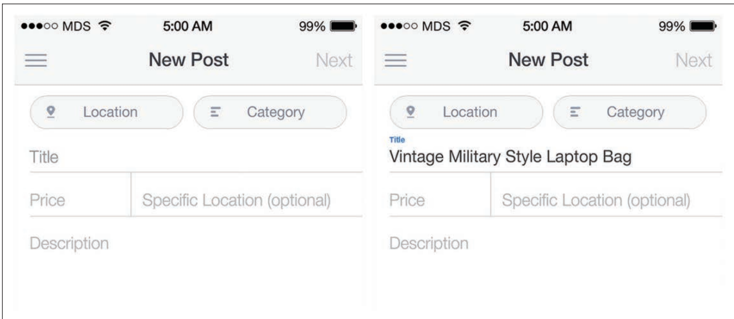



图 2-18

采用悬浮标签模式，只要输入文字，水印标签就会“悬浮”在输入框上方

还有一种不错的做法是，在适当的时候以内联的形式给予实时反馈，比如 Instagram 的账号名称设计。想了解更多技巧，请参阅第 8 章。

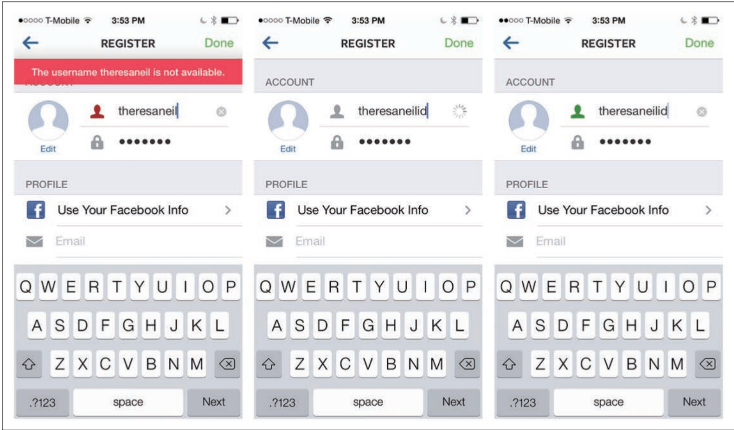



图 2-19

iOS 版 Instagram：内联形式的实时反馈

你是否会对密码的安全性有特定的要求，诸如字母的大小写等？如果有，那就一定要在用户点击“提交”或“完成”按钮之前告诉他们，不要在之后再显示错误提示信息。HelloWallet 的说明有些多，Level 相对而言则更简洁明了（尽管在我看来，它的总体注册体验还有改进的空间）。

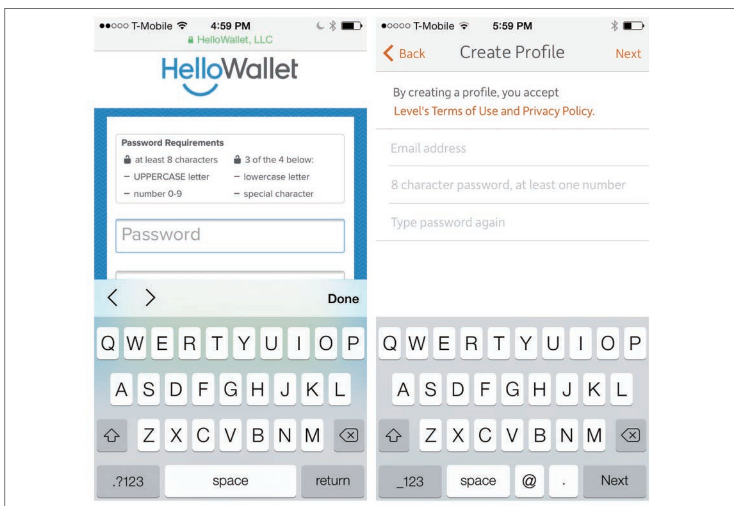



图 2-20
iOS 版 HelloWallet 和 Level：预先显示密码要求

一旦用户提交了注册表单，要立即给予注册反馈，参见安卓版 Etsy 和 Pageonce 应用。

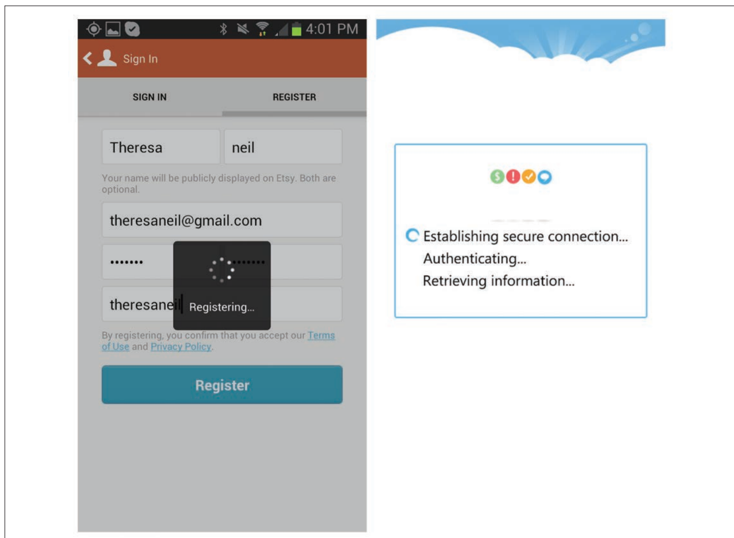



图 2-21
安卓版 Etsy 和 Pageonce：表单提交后就给予用户注册反馈

在注册过程中，可以的话要让用户始终停留在应用中。如果你必须链接到一个网页表单，至少在移动浏览器中测试一下以确保它的可用性。Windows Phone 版 Fitbit 的设计师显然没有进行这项工作。

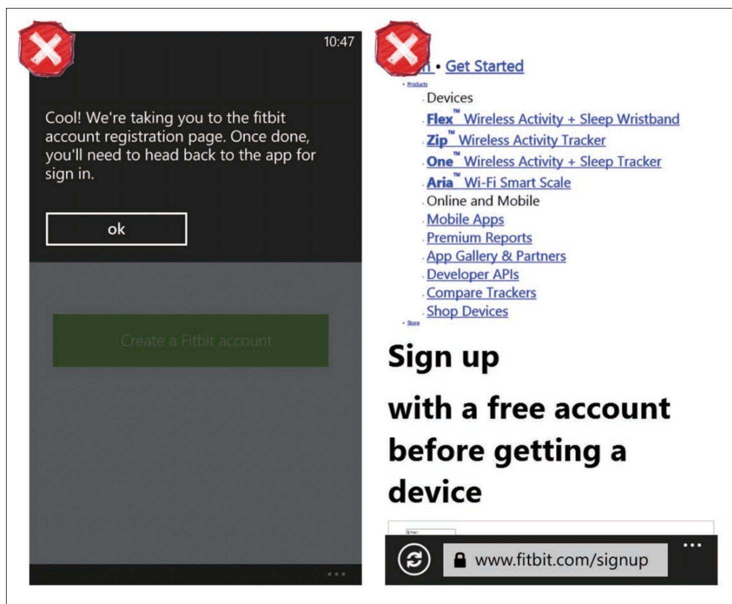



图 2-22
Windows Phone 版 Fitbit：令人痛苦的网页端注册体验

应该把注册当作一个流程，对用户输入的所有可能性考虑周全。再以 Level 为例，我已经有了一个账号，“不小心”又重新填写了注册表单。尽管在我的电子邮件地址旁边有一个对勾——这显然向我表明这个邮箱是可用的——但是我却收到了错误提示，原因是此邮箱地址已被使用。我只能点击“烦人”按钮（Bummer）关掉弹窗。

为什么不能在我填写表单字段的时候就告诉我电子邮件地址已被使用了呢？或者至少转到填好我的邮箱地址的登录界面，而不是给我一个硬生生的错误提示，且没有任何线索指明我下一步该做什么？

［小贴士］

注册表单要简短，表单长度最好在一屏之内，操作按钮要放在可视范围内。清楚地说明输入要求，可利用实时反馈加快注册过程，并让用户了解当前状况。要考虑到所有用户的使用情况。

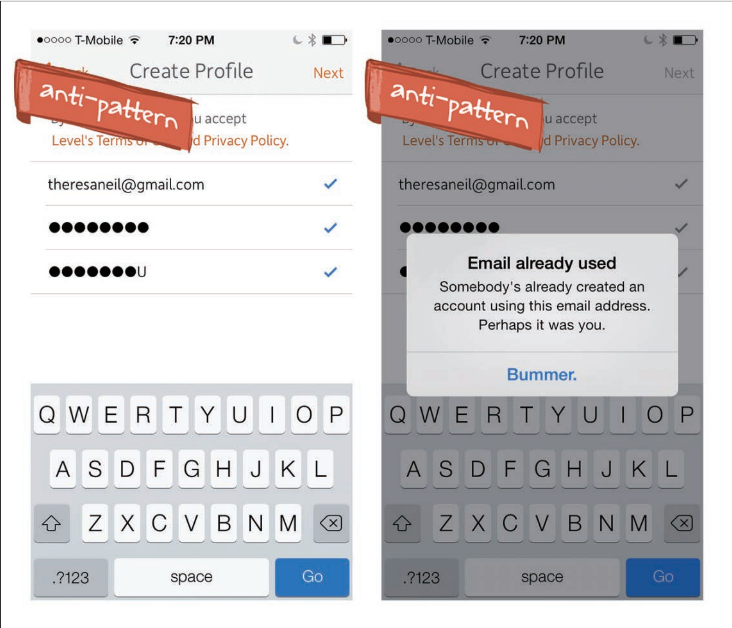



图 2-23

iOS 版 Level：我的电子邮件地址已经被占用，但旁边还显示了对勾

Dropbox 针对同样的情况给出了漂亮的解决方案。一旦识别出我的邮箱已存在，它会推测我是想登录而不是注册。因为我很可能需要找回密码的指引，所以它会预先在找回密码的页面填好我的邮箱地址，这样我只需点击“发送”即可等待恢复账号了。

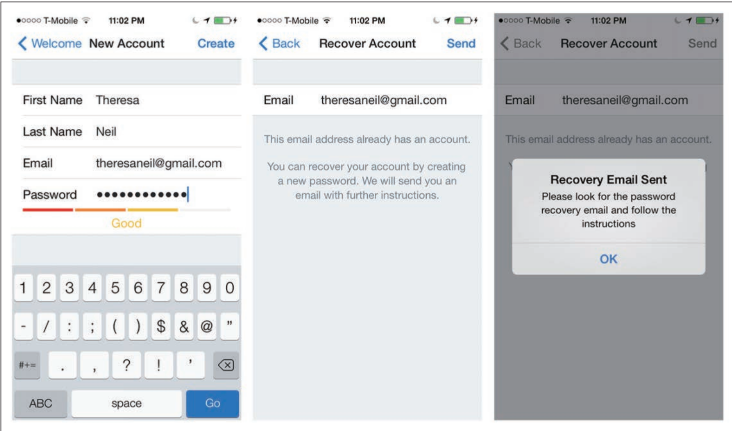



图 2-24

iOS 版 Dropbox：重复注册情况下聪明的解决方案

## 个性化注册表单

这是一种相对新颖的设计模式，其目的是为用户提供更加个性化的内容（如 Ness 和 News360），通过关联社交媒体提升用户体验（如 Path 和 Instagram），或是获取设置所需数据（如 Personal Capital）。

稍后将看到的多步骤模式为我们提供了更详细的注册流程方面的例子。

Facebook 最近发布的 Paper 应用以其拖放选择感兴趣的主题这一 UI 设计，成了“不必要的复杂”类型的反模式案例（参见第 11 章）。Ness 和 News360 则采用了滑动和点击的设计，这种设计更高率，而且很容易单指操作。若想进一步了解这一理念，可以参阅 Luke Wroblewski 的文章“Designing for Thumb Flow”（http://www.lukew.com/ff/entry.asp%3F1734%3D）。

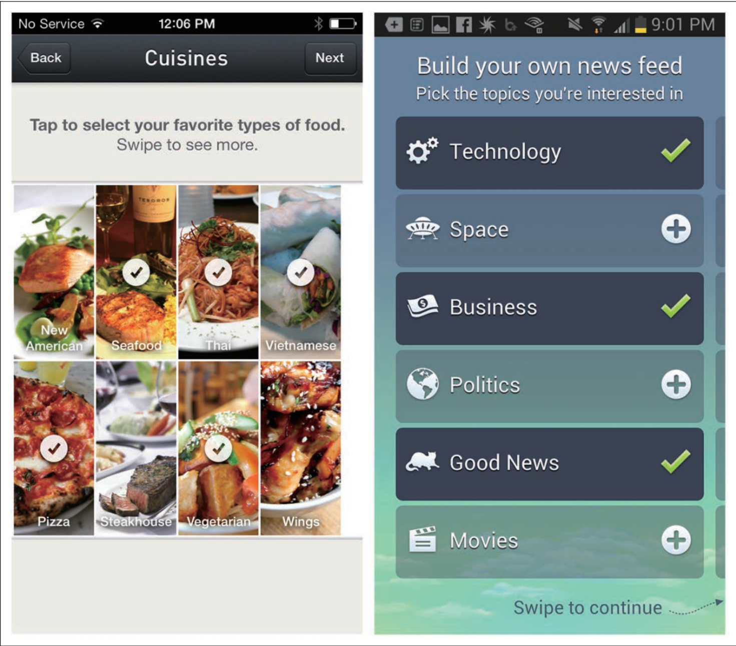



图 2-25
iOS 版 Ness 和安卓版 News360：注册流程中的偏好设置提示

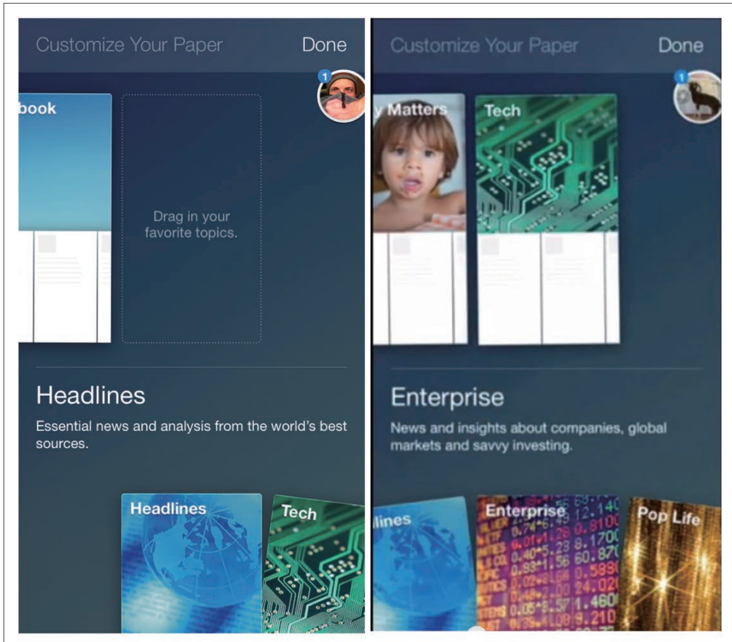



图 2-26
Facebook 出品的 iOS 版 Paper 应用：拖放这种 UI 设计会耗费用户更多的精力（http://youtu.be/UpZFw2wQr58）

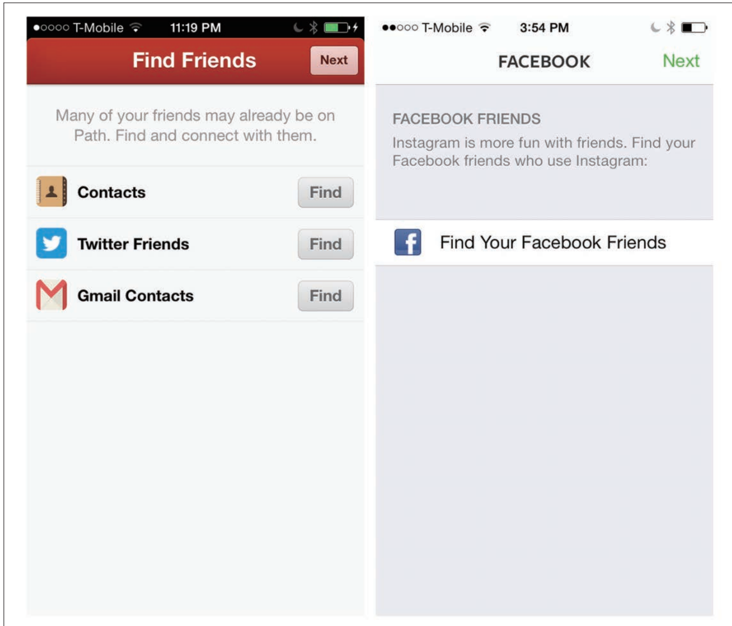
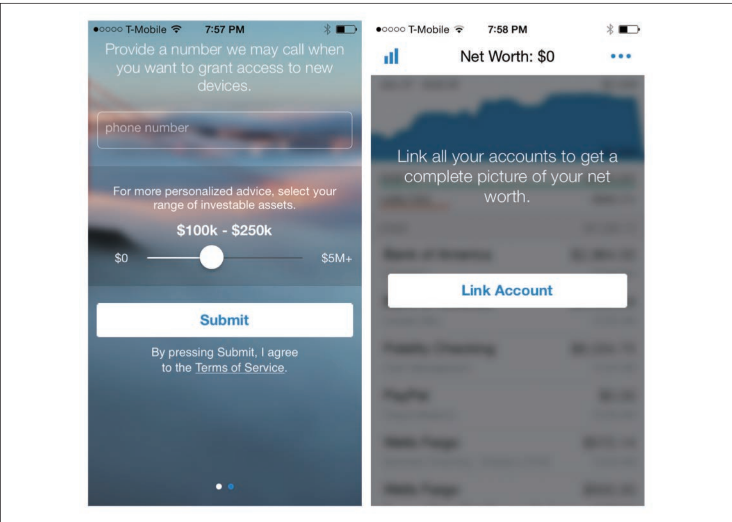



图 2-27
iOS 版 Path 和 Instagram：在注册流程中鼓励用户找到好友

图 2-28

iOS 版 Personal Capital：电话号码和资产范围的提示，同时提供链接账号的选项




## 多步骤表单

在网页端，多步骤注册过程经常使用进程栏（process bar）[1] 指示用户他们所处的位置，并提示下一步操作。但是移动设备的屏幕太小，没有足够的空间显示那么大的进程栏。你或许可以在标题栏里挤进一个，但是还有更好的方法来引导移动用户完成一系列步骤。

在 iOS 版 PayPal 的早期版本中，利益相关者坚持采用进程栏的设计。毕竟转账是一个流程。但现在的 PayPal 与其 2010-2011 版本相比，简洁度大大提高了。

图 2-29

大约 2010 年到 2011 年的 iOS 版 PayPal：转账进程栏




注 1：processs bar 译为“进程栏”，与进度条（progress bar）、进程条区别。（译者注）

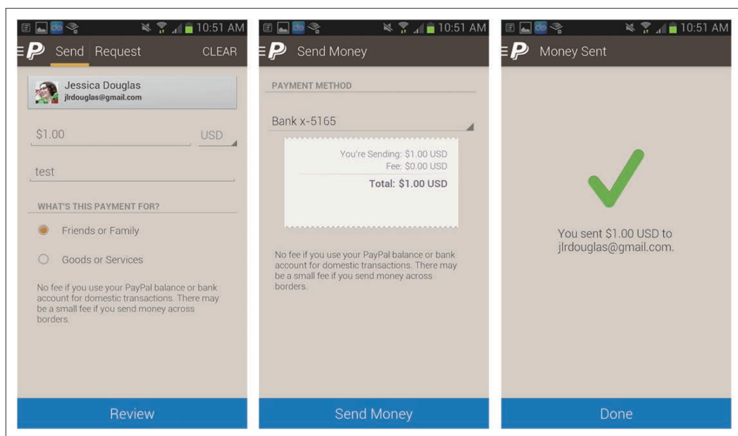



图 2-30
2013 年左右的 PayPal：整体看上去清新许多，原因之一就是去掉了进程栏

如果几个步骤不得不跨屏幕显示，那么你可以不采用过于密集的进程栏，而是采用显示当前步骤是第几步的方法，比如 SnipSnap（显示在标题栏中）和 Priceline（显示在页脚）。

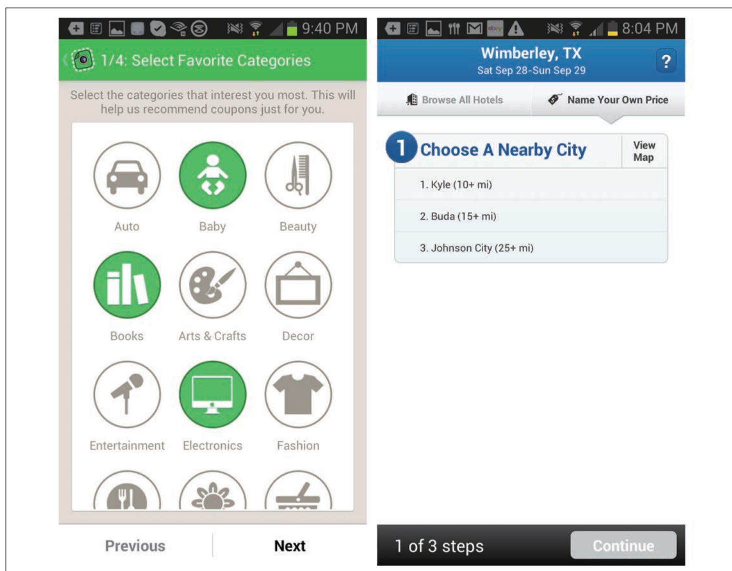
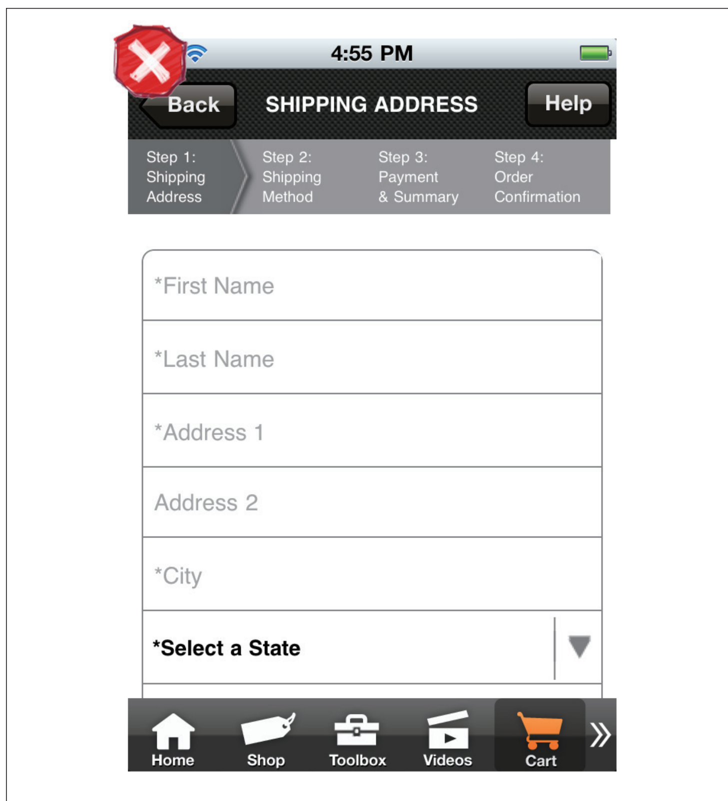



图 2-31
安卓版 SnipSnap 和 Priceline：表明当前步骤是第几步

在本书第 1 版中，Home Depot 的结账步骤被收入反模式案例中。它的进程栏上挤满了超小的文字，且结账体验冗长乏味。但现在这个应用的设计经过了大幅改进，结账页面改成了单一页面。

图 2-32
2011 年 的 iOS 版 Home Depot：过度拥挤的进程栏




安全结账过程的第一步是获取收货地址。当点击“下一步”时，你需要选择送货时间。虽然这时仍然显示收货地址，但处于只读模式，地址旁边放有“编辑”按钮。

选定送货时间后，点击“下一步”显示付款方式表单。完成付款方式表单后，你可以查看总数并核对此前输入的所有信息——但不要离开此页面——然后进入下一步。

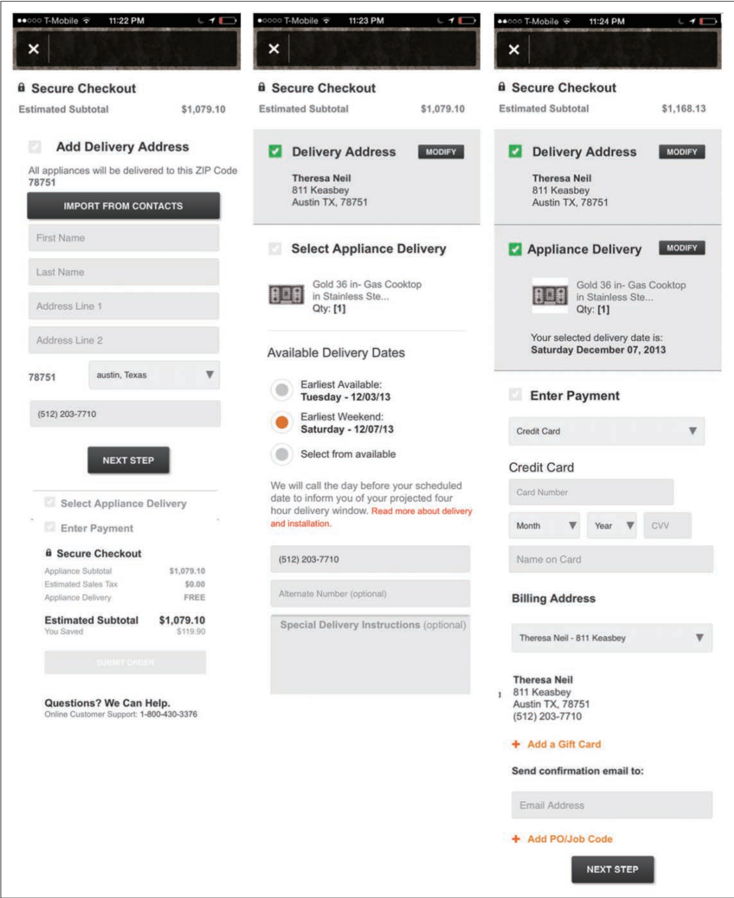
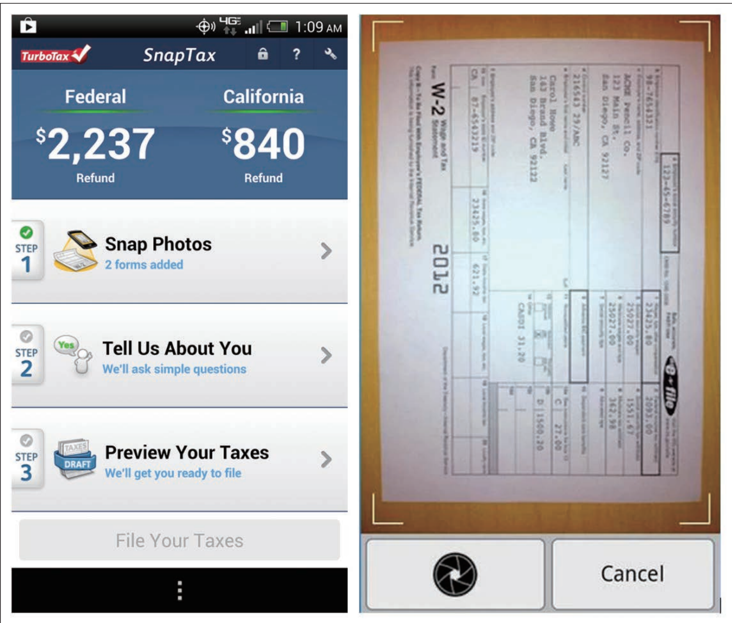



图 2-33
2013 年 的 iOS 版 Home Depot：大幅改进的结账流程

TurboTax SnapTax 采用了类似的设计。但是每一步的详情收起，可点击向下展开，而不是全部显示在同一页上。这个单页设计提供了有价值的导航信息（整个过程有三步，其中有两步还没有完成），同时也使高层级的摘要内容成倍增加。

图 2-34

Intuit 公司的安卓版和 iOS 版 TurboTax SnapTax：向下展开显示详情




［小贴士］

向用户显示他们当前所处的位置和下一步，但不要使用拥挤的进程栏。去掉不必要的输入字段，最小化页面的数量和步骤数量。

产品配置是一种特殊的多步骤表单，它不只是典型的线性流程，而是需要实现返回－前进的导航。星巴克为了在它的 Drink Builder 工具中实现这一效果，只使用了“下一步”和“返回”风格的按钮。[1]

这个表单用起来还不错，但是如果不打开每一个选项并滚动整个列表，就很难看出定制页上已经设置过的内容有哪些。

注 1：最新版本的美国区星巴克应用已经去掉了 Drink Builder 功能。此前用户通过应用定制的咖啡可以截屏保存成图片，并在购买时出示。（译者注）

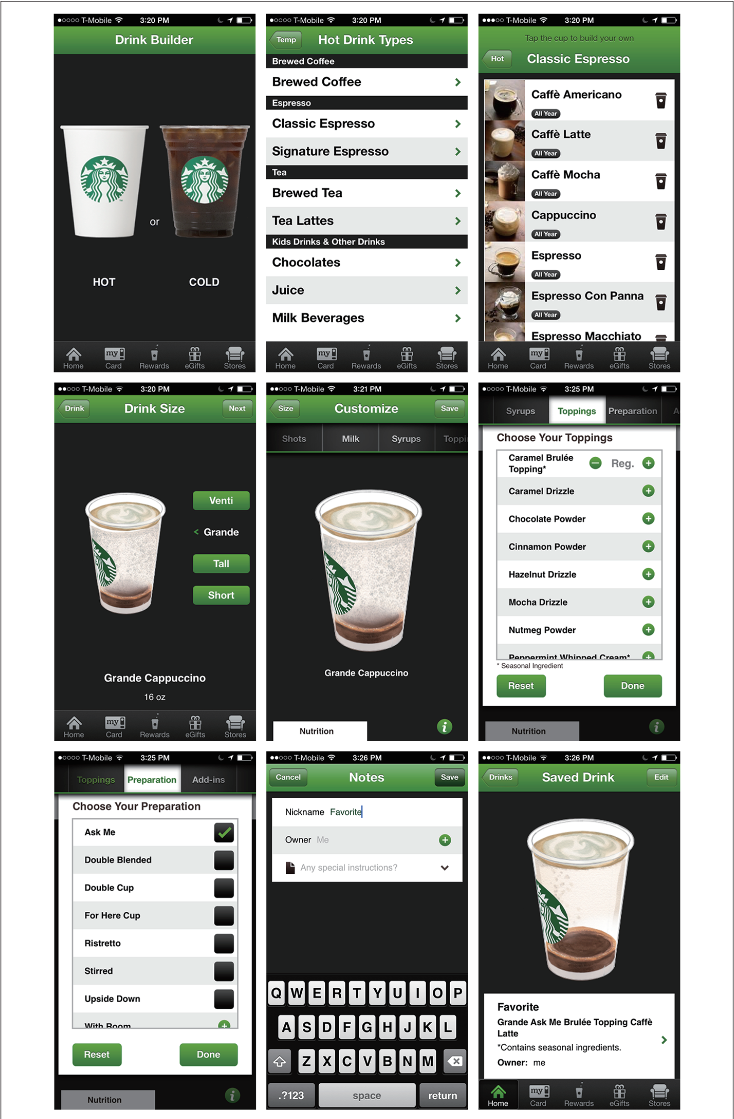
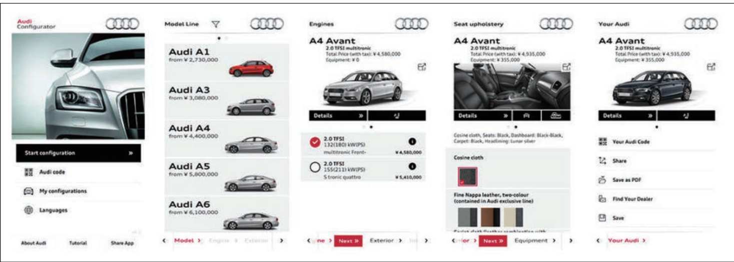



图 2-35

星巴克的 iOS 版 Drink Builder：“下一步”和“返回”风格的按钮（http://www.youtube.com/watch?v=Bvke0dXD2SM）

一个简单的修正办法是，在底端的“成分”（Nutrition）标签和“信息”（Info）图标附近固定放置一个“概要”（Summary）或“详情”（Details）按钮。Audi Configurator 就完完全全是这样设计的，点击“详情”按钮打开显示目前选择的全部参数。

图 2-36
iOS 版 Audi Configurator：点击固定的“详情”（Details）按钮显示选定的选项

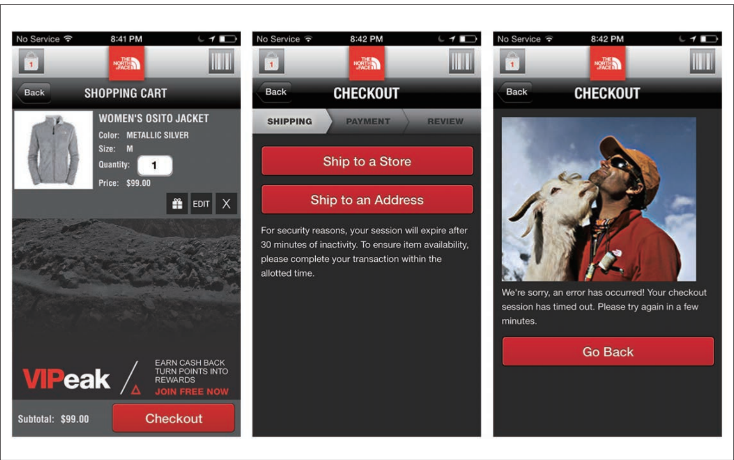



## 结账表单

截止到 2013 年 1 月，有大约 15% 的电子商务是在移动设备上进行的，几年内这一数字有望超过 25%。然而，即便是从手机端开始的购物体验，大多数人也会选择在桌面端、平板或是零售商那里完成交易或结账。

在许多案例中，这种情况的出现多半要归咎于极为糟糕的结账体验，这在众多商家的原生应用中屡见不鲜。换言之，这是因为许多商家都只让用户在其应用中购物，而要跳转至优化后的移动网站进行结账。

图 2-37
iOS 版 The North Face：每当我要结账时都显示连接超时！呃！为什么要出现连接超时呢？




让你的用户在兴致最高的时候就能购买。不要让差劲的用户体验阻碍你的客户在手机上结账，最好让他们就在购物的应用里结账。下面是一些高转化率的支付流程的设计技巧。

## 技巧1：同时提供登录、注册和客人用户选项

允许客户在账号信息中保存其个人资料，以便将来快速结账，还应提供客人用户结账的选项。你完全可以在用户完成支付后提示他们去创建账号，这样给人的体验会更舒服。

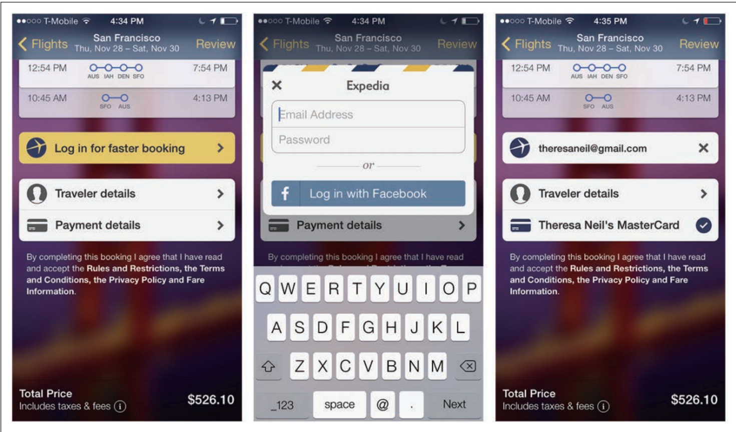



图 2-38
iOS 版 Expedia：登录之后填写旅客结账表单与付款细节

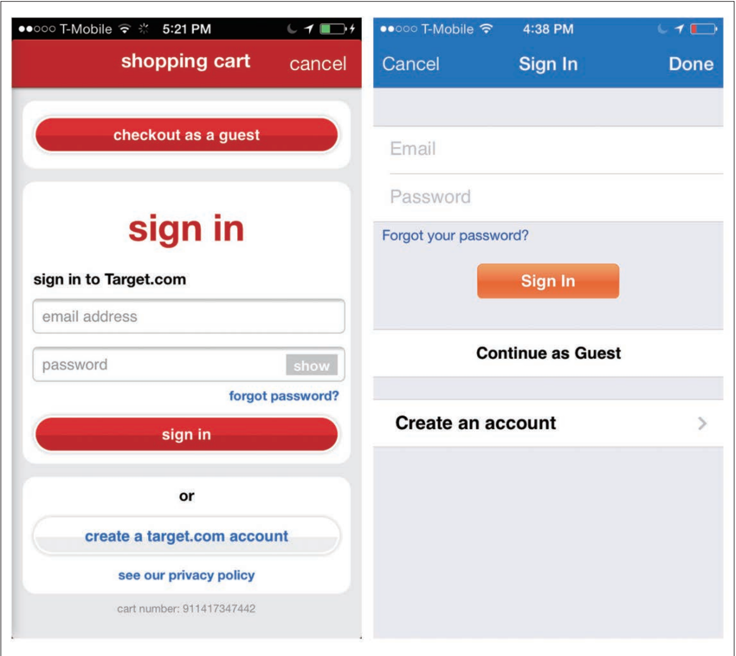



图 2-39
iOS 版 Target 和 Walmart：既提供了登录 / 注册选项，也提供了客人用户结账的选项

## 技巧2：简化流程

弃用进程栏，把分屏步骤融合进一个结账页（参见“多步骤表单”一节中关于 Home Depot 这一优秀例子的讨论）。Haute 和苹果公司等零售商设计了包含多个区块的单一结账页表单，方便用户逐级深入地点击，直到完成支付。

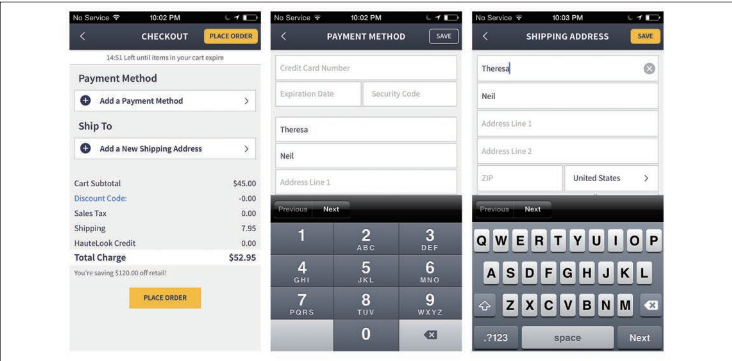



图 2-40
iOS 版 Haute：单一结账页上有额外的页面，用于添加详细信息

## 技巧3：提供省时的快捷方式

提供“导入通讯录地址”选项，以便用户能够快速填写收货地址和账单地址。还可以考虑采用“扫描卡片”的功能，为用户省去输入全部信用卡信息的麻烦。

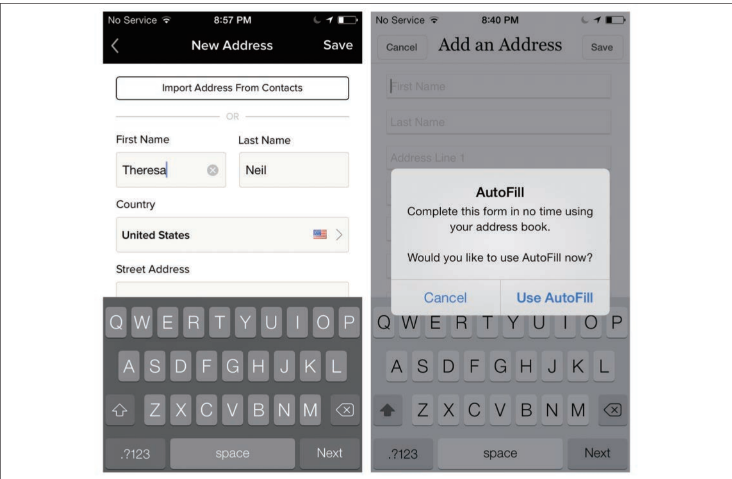



图 2-41
iOS 版 Gilt 和 RuLaLa：输入地址时可选导入 / 自动填充

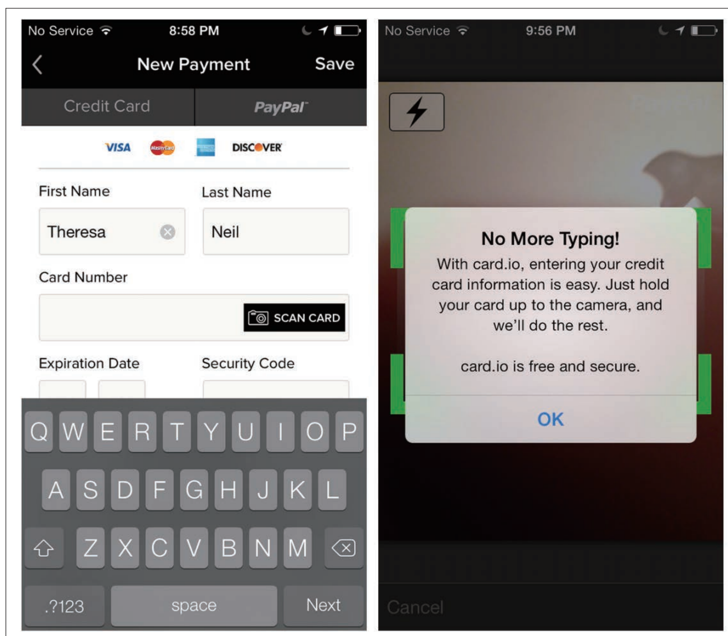



图 2-42
iOS 版 Gilt：由 card.io 公司提供支持的“扫描卡片”选项

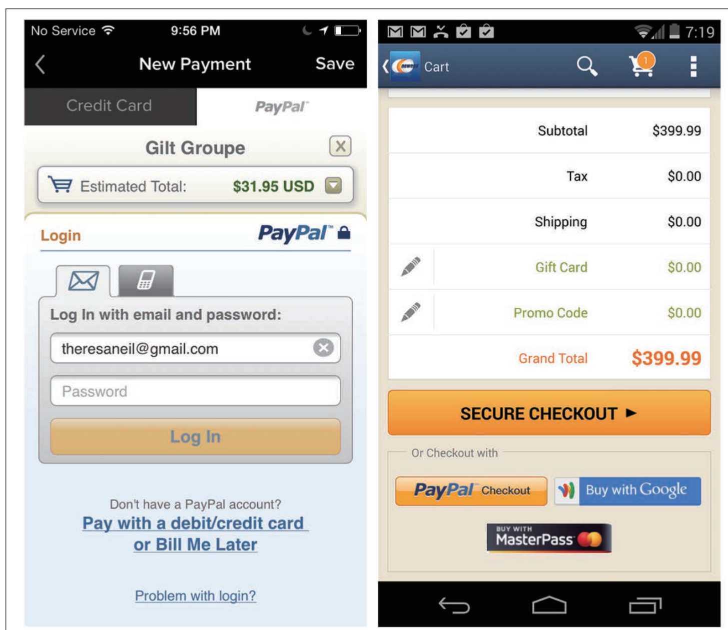
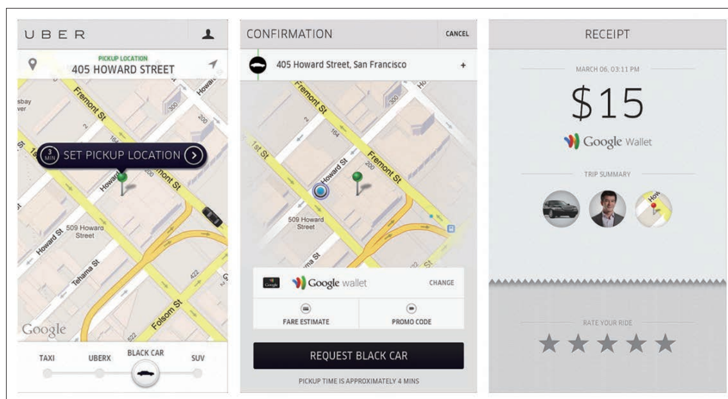



图 2-43
iOS 版 Gilt 和安卓版 Newegg：其他可选的付款方式，通过 PayPal 和 Google Wallet 付款

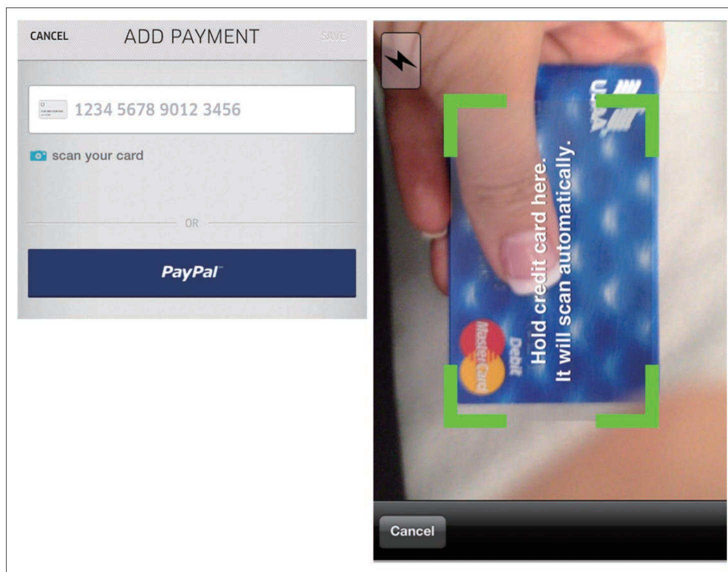
图 2-44
iOS 版 Uber 集成了 Google Wallet，简化了订车流程




## 技巧4：提供快速结账方式

考虑为注册用户提供一种极度快捷的结账流程。苹果商店提供了一个快速结账（Express Checkout）的选项，让用户可以自行设置。亚马逊则在产品详情页上放置了一键结账按钮。在下面的截图中，有两个移动端的一键结账按钮，都使用了色彩鲜艳的、行为召唤的橙色和形象的一键点击图标。

图 2-45
iOS 版 Uber 也提供信用卡扫描功能和使用 PayPal 付款的选项



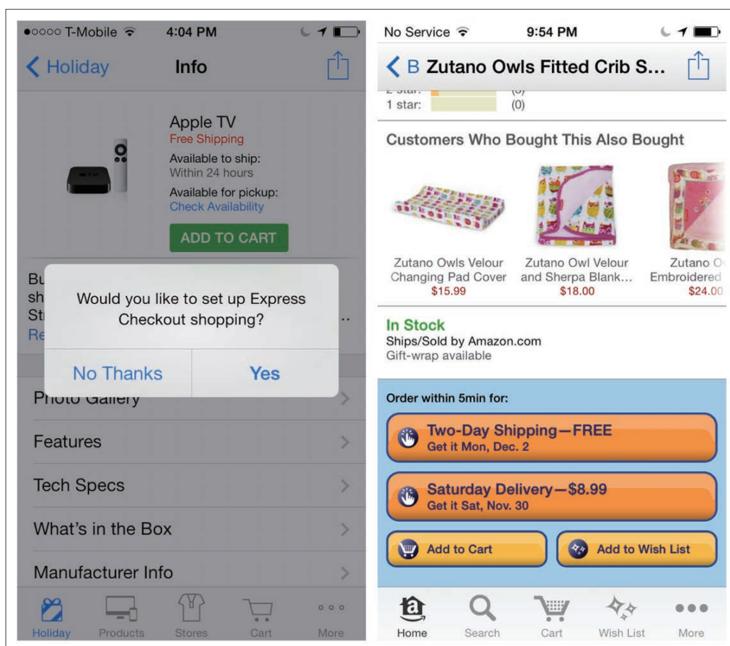



图 2-46

iOS 版苹果商店和亚马逊：快速结账选项

## 技巧5：忘掉网页端

我们已经能自如地在网页端购物，购物车的隐喻和多步骤支付流程对我们来说已经相当熟悉了，但移动零售还有很大的创新空间。苹果、沃尔玛、Target 以及其他零售商都着眼于更多的移动端销售转化，不断地在店内店外尝试新的结账体验。

比如，沃尔玛的 Scan & Go 功能让店内消费者可以通过扫描商品的方式来将其加入购物车，这避免了在店内排队结账。据沃尔玛称，超过一半的新用户最终都会再次使用该功能。

[小贴士]

通过简化支付流程、去除不必要的步骤并提供快捷方式，完成移动端转化设计。观察过去的网页端购物流程，同时充分利用原生移动设备技术。

图 2-47

iOS 版沃尔玛：Scan & Go 功能让你可以扫描选购的商品，省去排队结账的麻烦

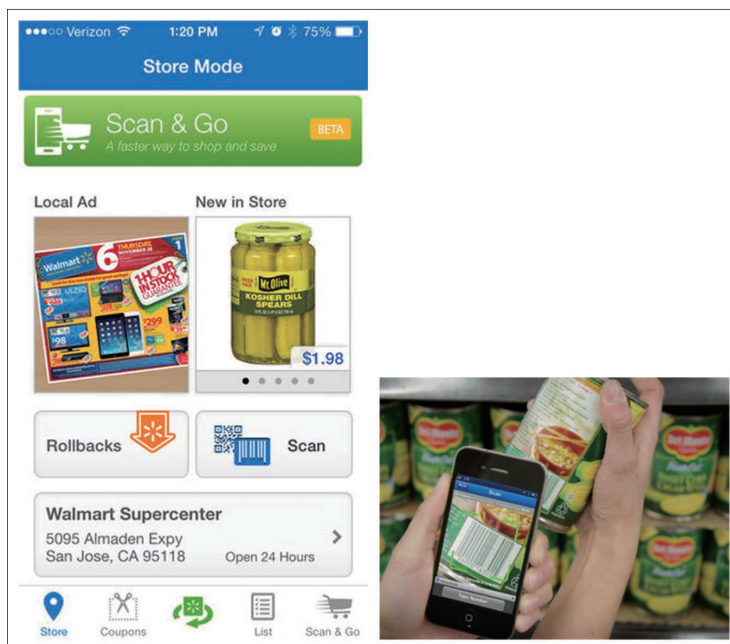



图 2-48

苹果商店拥有 EasyPay 自助结账功能

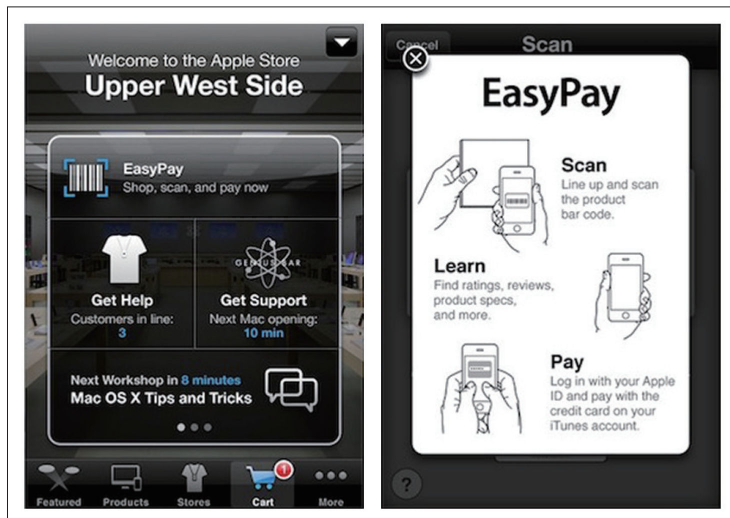





图 2-49
沃尔玛加拿大联合美泰（Mattel）设立的虚拟玩具店（http://www.newswire.ca/en/story/1253739/mattel-and-walmart-canada-offer-holiday-shoppers-toys-on-the-go）：扫描二维码购买商品




图 2-50
Target 和 CBS Outdoors 合作(http://www.cbsoutdoor.com/creative/coolcampaigns/target-holiday-gift-finder.aspx)：公交车站成为“数字假日体验”

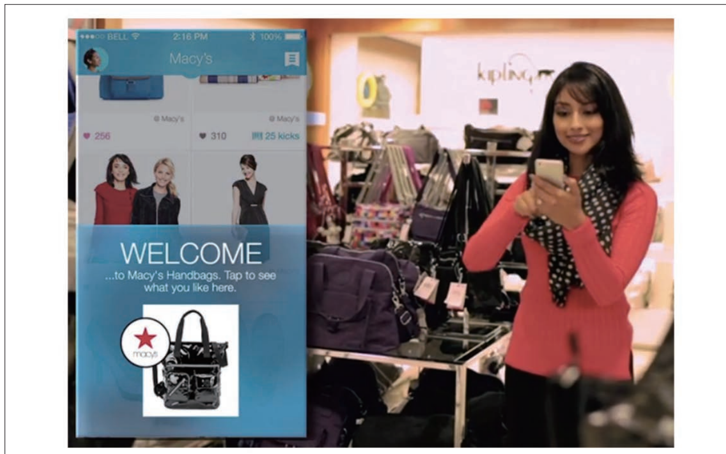



图 2-51
ShopKick 用来找优惠的 Shop Beacon（http://techcrunch.com/2013/11/20/shopkick-rolls-out-ibeacon-enabled-shopping-alerts-with-shopbeacon-with-macys-as-its-first-trialist/）和用来接收付款的 PayPal Beacon（https://www.paypal.com/webapps/mpp/beacon），都使用了基于附近位置的蓝牙技术，个性化并简化了移动购物

## 计算表单

计算器风格的应用，比如体重跟踪器、税款估算器以及贷款计算器，都需要用户输入。虽然这些表单可以采用自定义控件和布局，但是遵循基本的设计规范以保证表单的易读性依然十分重要。

对齐方式、标签、字体、按钮位置、对比及颜色，这些全部都是影响移动表单易用性的因素。比如，比较一下输入字段和标签设计极为优雅的 Valspar Paint Calculator 和 Behr Paint Calculator 两者的易读性。

图 2-52

Valspar Paint Calcu lator 的输入字段整齐优雅，比 Behr Paint Calculator 更易于阅读




最好的计算器应用应该将输入结果用视觉效果呈现出来。TaxCaster 使用了测量仪风格的图表，将应缴税款和应退税款可视化。测量仪在数据输入后即时更新。

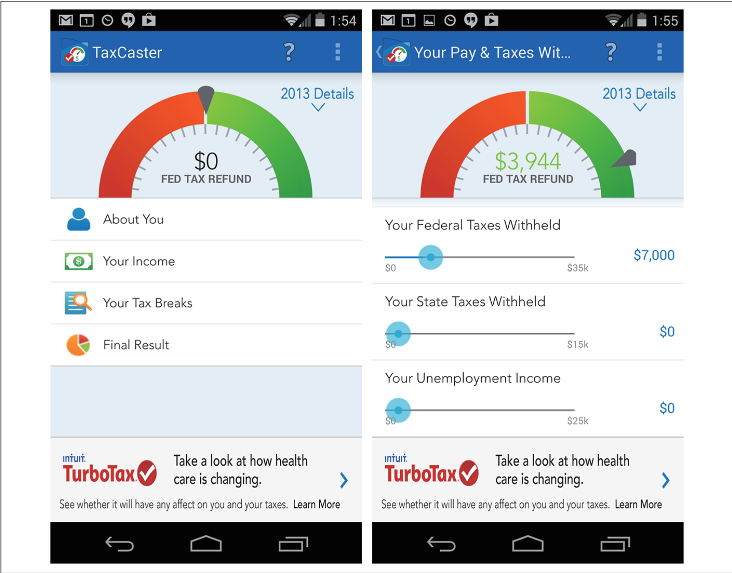



图 2-53
安卓版 TaxCaster：输入变化时测量仪随之更新

Zillow 和 Trulia Real Estate Calculators 也提供了输入数据的实时可视化。

［小贴士］

计算器表单布局应该使用标准化表单设计。在符合应用使用场景的情况下，增加交互可视化。

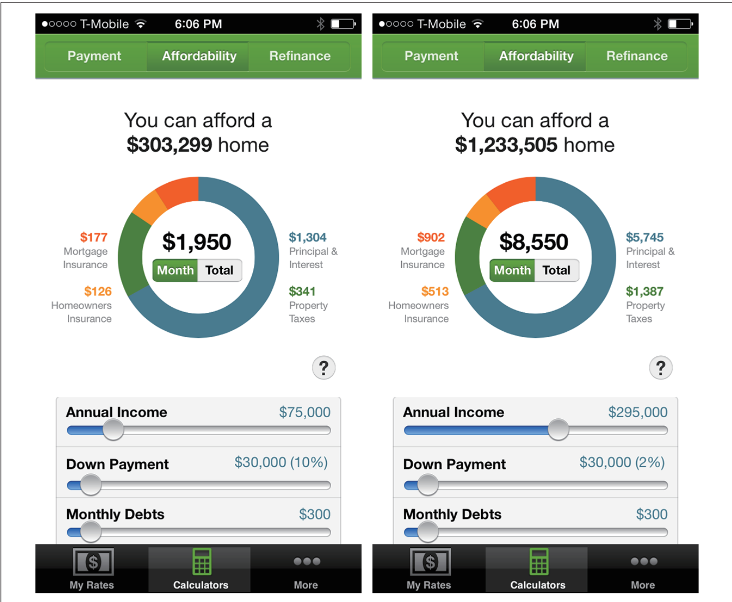
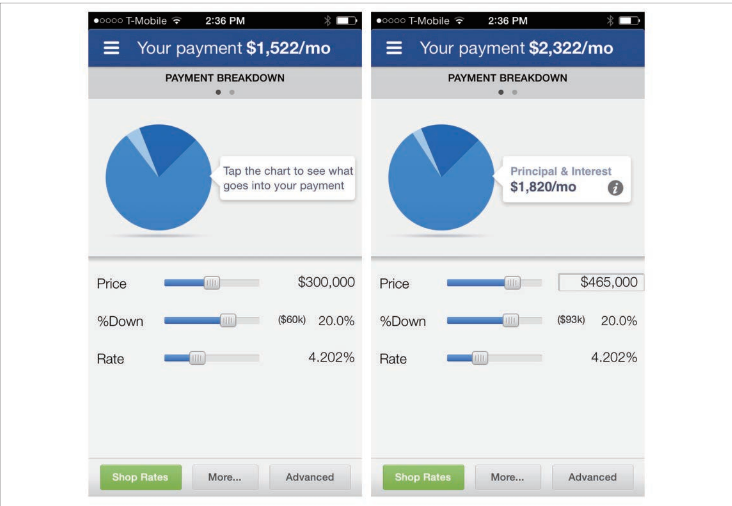



图 2-54
iOS 版 Trulia Real Estate Calculators：实时反馈

图 2-55

iOS 版 Zillow Mortgage Calculator：输入数据时显示变化



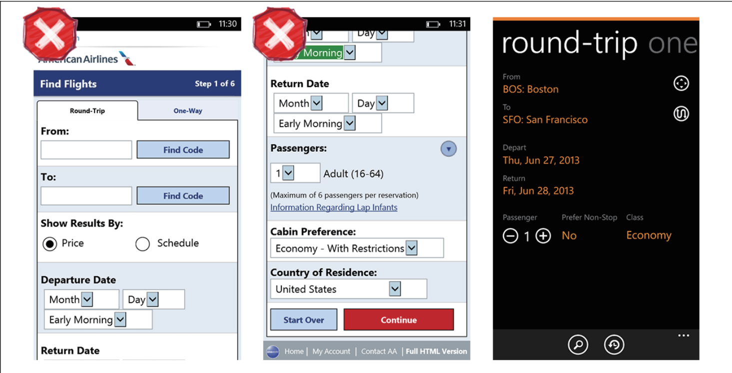

# 搜索表单

有些搜索功能需要输入多种条件以生成搜索结果。和其他表单模式一样，搜索表单应该只有最核心的、不可或缺的输入字段，并提供合理的默认值。你完全可以在搜索结果页提供筛选功能，让用户对结果列表进行细化。在这里我只讨论表单本身，关于搜索、排序和筛选模式的更多内容，可参阅第 4 章。

比较看看 American Airlines 搜索表单的混乱布局与 Kayak 的清爽设计。

图 2-56

2013 年 Windows Phone 版 American Airlines 和 Kayak：凌乱与整洁



Zillow 在表单页上可直接预览搜索结果的数目，这个设计不错。如果你注册了 Zillow 账号，可以保存这项搜索，为将来的搜索节省时间和精力。

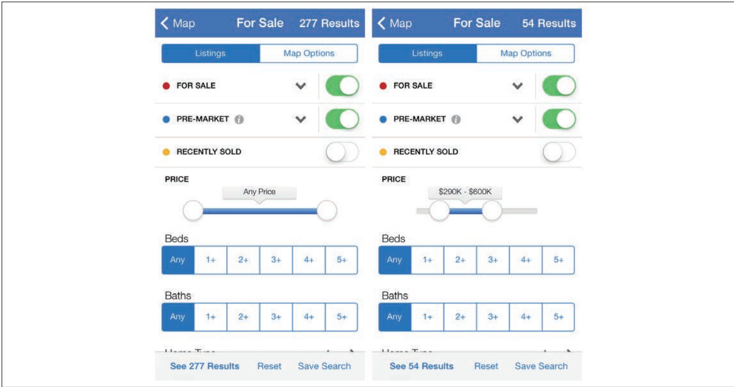



图 2-57
iOS 版 Zillow 可以在输入表单时实时显示搜索结果的数目

Hipmunk 的搜索表单同样简洁，而且还增加了一个票价提醒功能。票价提醒和保存搜索功能相似，都是通过保存搜索条件进一步提高移动用户的效率；与简单地保存搜索不同的是，当票价变化时，票价提醒功能可实现应用内通知。这是 Hipmunk 获得成功的关键因素之一，因为这个提醒使用户一次又一次重新打开应用。

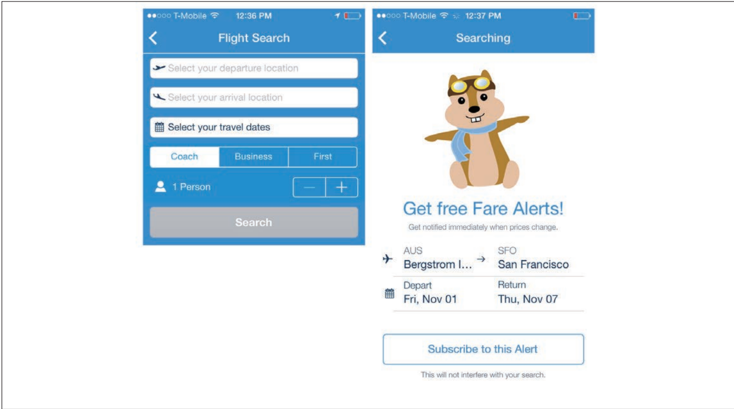



图 2-58
iOS 版 Hipmunk：票价提醒功能为保存搜索功能添加了省钱利器

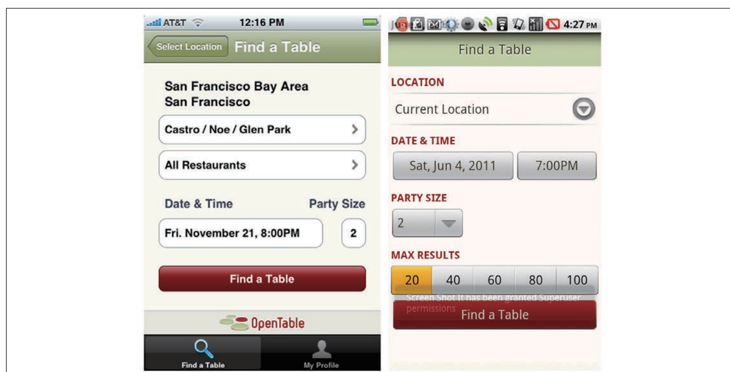
让我们看看 OpenTable 2011 年的搜索表单，再对比一下它为 iOS 7 新设计的表单。早期的设计版本是一个完整的搜索表单，有 4 到 5 栏输入字段，还有一个醒目的红色行为召唤按钮。

[小贴士]

搜索表单要短而简单，搜索条件尽量控制在一页内或更少。提供合理的默认值和保存搜索的选项，以便将来使用。

图 2-59

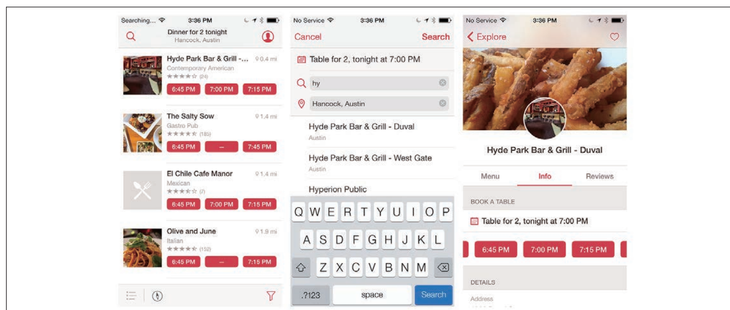
2011 年 OpenTable 的 iOS 版和安卓版：用户必须输入全部搜索条件




在 iOS 7 版的设计中，首页是附近的餐厅列表和当天可能订到座位的时间。这是一个隐式搜索（Implicit Search）的例子，我们会在第 4 章中讨论。现在的搜索表单上只有三个输入字段，其中餐厅名称一项还可开启自动建议功能。

图 2-60

iOS 7 版 OpenTable：隐式搜索自动建议附近可选的餐厅




# 长表单

一些表单不能只放进一屏之内。在这种情况下，长的滚动页比把表格切分成多页要好得多。设计长表单最棘手的部分在于，应在何处放置执行和退出按钮。

在大多数 iOS 应用中，表单在模态环境下显示。iOS 7 设计指南（https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/Modal.html#//apple_ref/doc/uid/TP40006556-CH64-SW1）中建议，命令按钮放在右上角，退出键（一般叫作返回或取消）放在左上角。

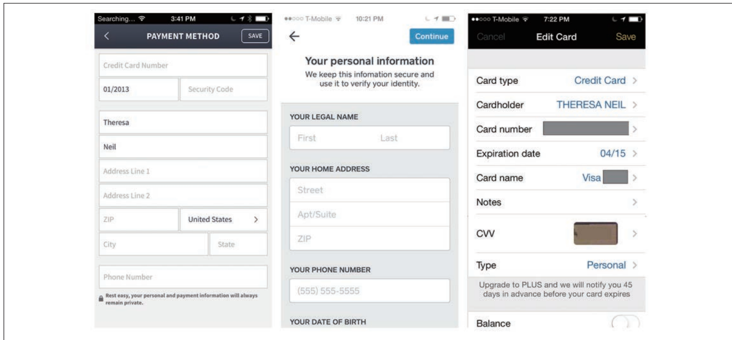



图 2-61
iOS 版 HauteLook、Square 和 Lemon Wallet：命令按钮放在右上角

Windows Phone 8 设计指南（http://msdn.microsoft.com/library/windows/apps/ff431806(v=vs.105).aspx）中建议，命令按钮应该放置于应用栏（固定显示在页面底端）上。

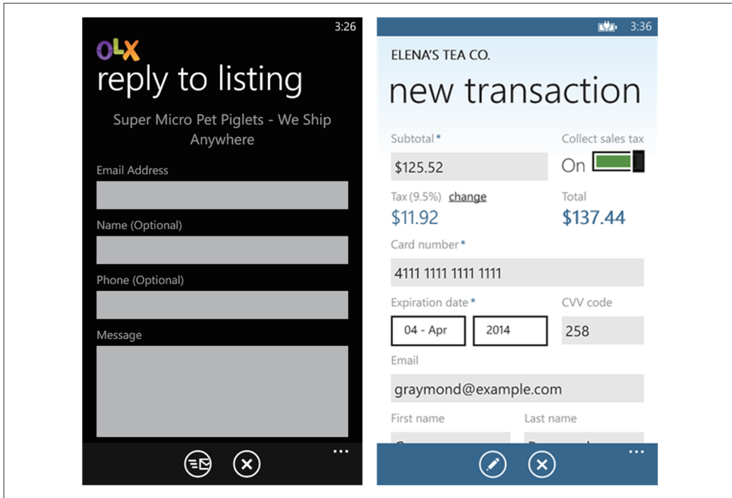



图 2-62
Windows Phone 版 OLX 和 Innerfence：命令按钮放在应用栏上

然而，安卓指南的建议（http://developer.android.com/design/patterns/actionbar.html）则完全不同。自从 Jelly Bean（安卓 4）发布以来，安卓的设计指南就建议，命令按钮应该放在操作栏（固定在屏幕顶端）上。

遗憾的是，我在 2013 年年末撰写本书的时候，这条安卓设计指南在测试中表现不佳。在我们对众多客户做的用户测试中，大部分用户找不到或者根本没看见操作栏上的按钮。随着越来越多的用

**［小贴士］**

不要特意把表单分割成多页步骤，单页滚屏更可取。要果断放弃任何没有必要的输入字段。按钮位置的设定要遵守操作系统规范。

户升级他们的设备和操作系统并对新的导航规则越来越熟悉，这一可用性问题可能会逐渐减少。与此同时，一些安卓应用还是会在底端位置固定显示命令按钮。

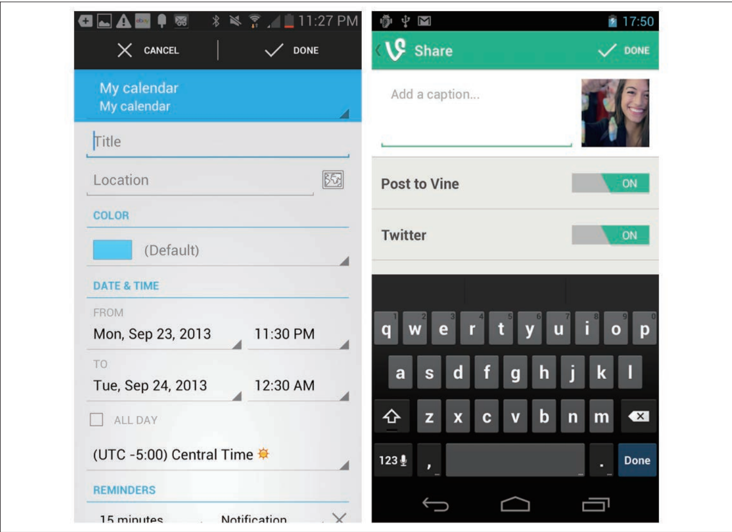



图 2-63

安卓版 DigiCal 和 Vine：模态表单和简洁的分享表单示例，它们分别将命令按钮放在了操作栏上

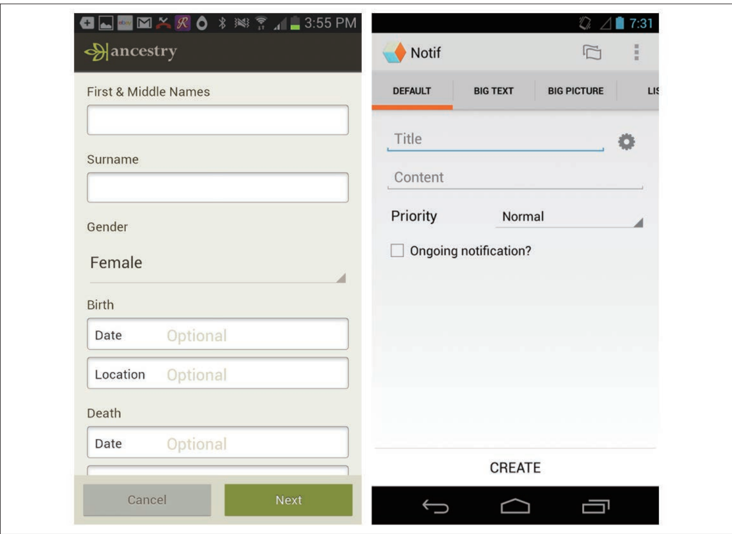



图 2-64

安卓版 Ancestry 和 Notif：命令按钮固定在底端

第 3 章

# 表格




- 常见表格模式

  基本表格（Basic Table）、无表头表格（Headerless Table）、固定列表格（Fixed Column）、概览加数据型表格（Overview plus Data）、行分组表格（Grouped Rows）、带有视觉指示器的表格（Tables with Visual Indicators）、可编辑表格（Editable Tables）

本书第 1 版中，我在本章开头提到，我公司的许多企业客户都需要用表格展示密集数据。由于对使用表格阅读数据非常熟悉，所以他们充满好奇，很想知道我们会怎样把那些表格搬到移动设备上。和现在一样，我的回答是“我们不打算那样做”。至少，我们不会不予说明移动设备本身的因素给设计师提出的要求。

在移动端显示表格数据是一个挑战，同时也是一个机会。挑战在于，要找到在小屏幕上显示数据的可用办法。而机会在于，我们可以更严苛地

审视数据，从而确定用户真正需要从中获得的是什么，然后研究出创新性的解决方案来显示数据，这样的表格甚至比传统表格更有用。

再来看看我在本书第 1 版中用过的例子——一张记录学生测验结果的长表格。这个表格用图表的形式可以表现出更好的效果，用户可以在钟形曲线图的分数区间上点击查看该区间内全部学生的分数。

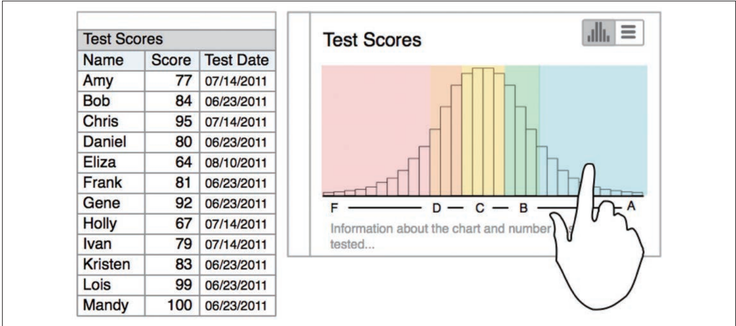



图 3-1
重新设计的表格：转换成一张交互式图表也许更有用

我们可以用另一种视图来展现这些信息，即利用具有动态搜索功能的无表头表格，用户可以快速查看特定学生的测验结果。

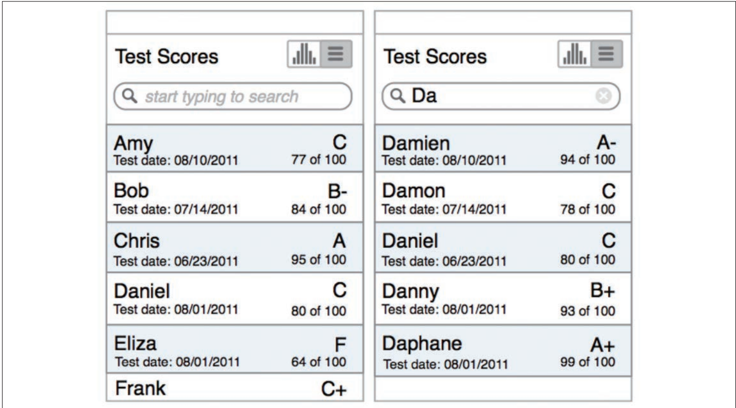



图 3-2
带有动态搜索的无表头表格：钟形曲线图的另一种表现形式

虽然这些模式在很大程度上和前几年没有什么差别，但设计师们已能更自如地运用它们来清晰而优雅地展示表格数据了。一旦确定了你的应用中需要显示的关键数据，就要了解下面这些表格模

式，从中找到灵感。

## 基本表格

基本表格模式就是带有固定列表头及网格布局的标准表格。设置不同的表格行颜色（这种形式也称为“斑马纹”），或者在两行之间用细线分隔，都能提高表格的可读性。

现在普遍省略垂直网格线以减少视觉噪音。这样做没有问题，但要保证列对齐。对比 ESPN SportsCenter 应用和 Stock Trainer 表格的可读性，就能看出列对齐的重要性。

[小贴士]

不要使用深色的网格线。如果不用垂直网格线的话，要确保列对齐。文字左对齐，数字右对齐。如果要在一屏内显示大量信息，请考虑采用其他的表格模式。

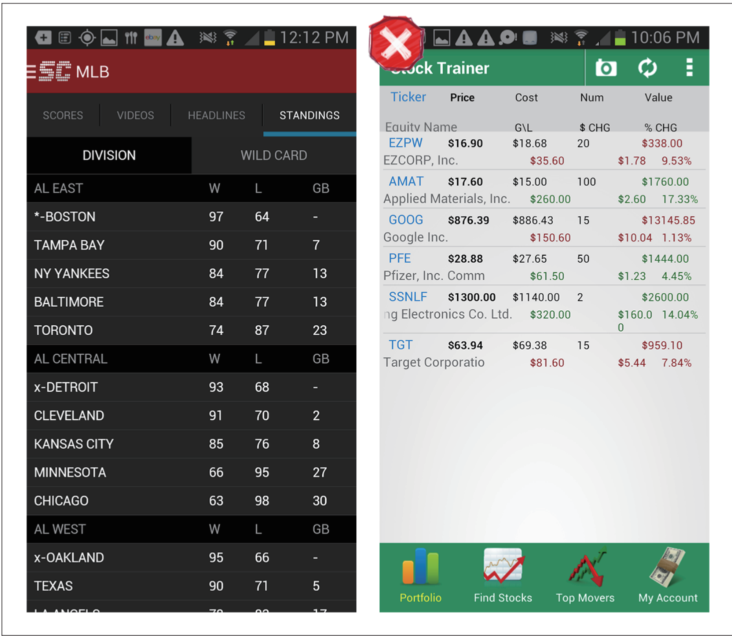



图 3-3

安卓版 ESPN Sports Center 和 Stock Trainer：如果去掉垂直网格线，就要保证列对齐

## 无表头表格

无表头表格的特征是，用较宽的行显示某一对象的多个变量，且没有列标签。常见的做法是突出显示行标识符（信息的关键部分）。BillGuard 采用了图标，而 Gas Guru 则使用了更大、颜色更深的字体来突出行标识符。

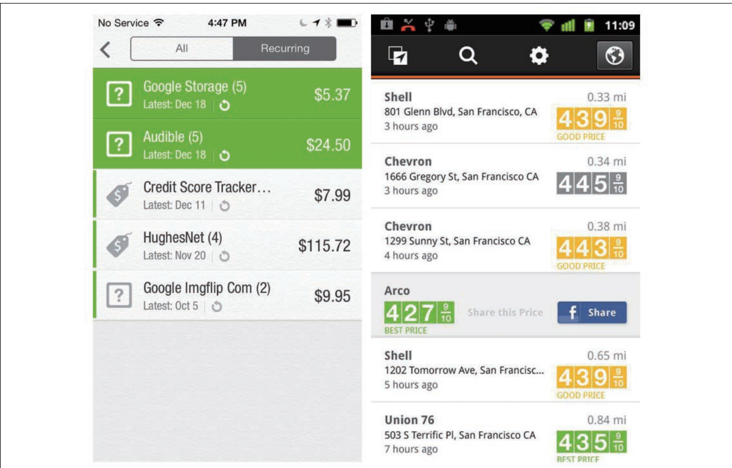



图 3-4

iOS 版 BillGuard 和安卓版 Gas Guru：醒目的行标识符

无表头表格非常适合用来显示项目集合（比如存货清单、菜谱、专辑等），也很适合显示搜索结果。FlightBoard 显示选定机场的航班信息。当用户点击一行时，航班详情会在行内显示出来，而不是跳转到新的页面。

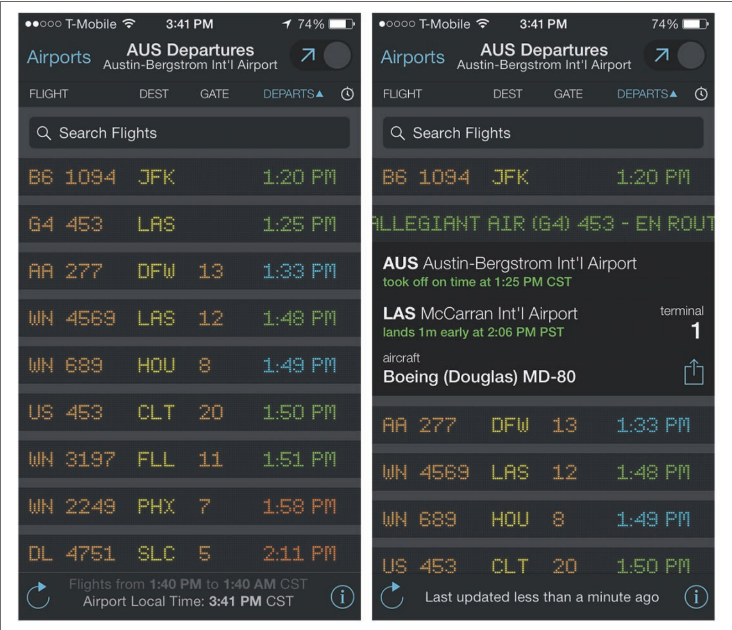



图 3-5

iOS 版 FlightBoard：无表头表格，点击行内展开详情

无表头表格和列表一样，都是为了快速浏览并执行操作。对齐、字体和颜色这些视觉设计元素极为重要。举例来说，Realtor.com 中的表格就比 Redfin 中的表格更易于浏览。

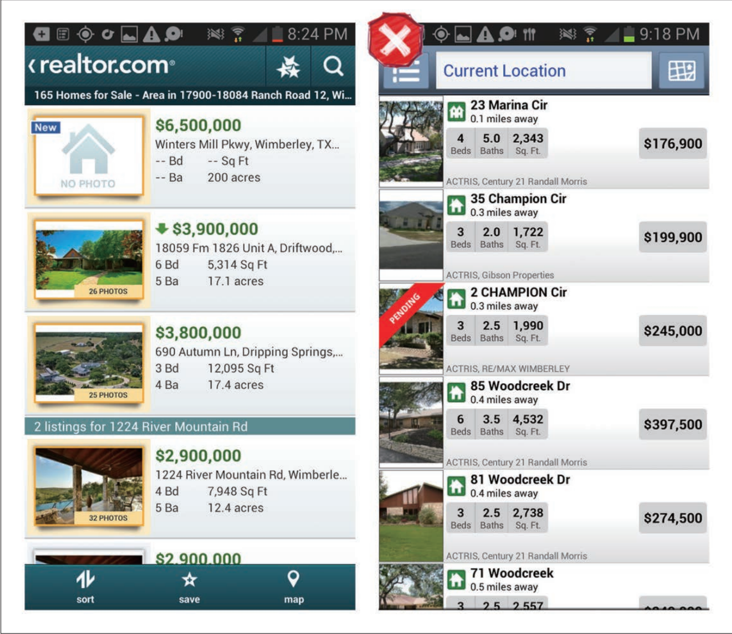



图 3-6
在安卓平台上对比 Realtor.com 和 Redfin：清晰的设计和美观的排列在无表头表格中极为重要

[小贴士]
去除噪音（图标和边框等多余的视觉元素），并设计精确对齐的表格以方便浏览。在不那么重要的细节上使用较小和颜色较浅的字体。不要猜测最重要的信息是什么，要征询客户的意见，然后通过测试确定该项设计。

## 固定列表格

对较大的表格来说，固定列表格模式或许是一个可行的解决方案。在 Roambi 和 Fidelity 的例子中，当用户不是垂直滚动屏幕的时候，最左侧的列固定，其他列可左右滑动。由于 Roambi 的超扁平化设计，乍看之下要分辨出左侧为固定列有点困难，但你可以看出右边有更多的信息。Fidelity 设计中的视觉功能可见性能让我们知道首列是固定的且按顺序排列。

[小贴士]
要提供视觉功能可见性，好让用户知道通过滑动可以显示更多的列。页面默认显示最重要的列。

图 3-7

iOS 版 Roambi：注意最左侧一列是固定的（用细微的阴影标识出来）http://epub.ituring.com.cn/Article/CheckIn/2434

图 3-8

iOS 版 Fidelity：功能可见性显示左列固定，通过滑动可显示更多列

# 概览加数据型表格

概览加数据型表格模式的特点是，表格内容的摘要显示在第一行数据的上方。这种模式在财务类型的应用中十分普遍，但也适用于任何其他应用，只要用户想从中看到总数或随时间变化的趋势。

概览可以是文字形式的，但最常见的是图表式的图像，请看下面这些例子。

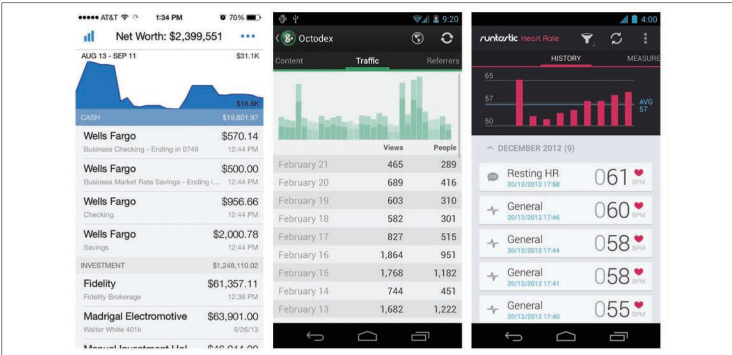



图 3-9

iOS 版 Personal Capital 和安卓版 Runtastic Heart Rate：概览加数据型表格

BillGuard 的概览图表是交互式的。如图所示，点击“VS. AUG”[1]，就会显示对比折线图，并标示出支出金额的变化。

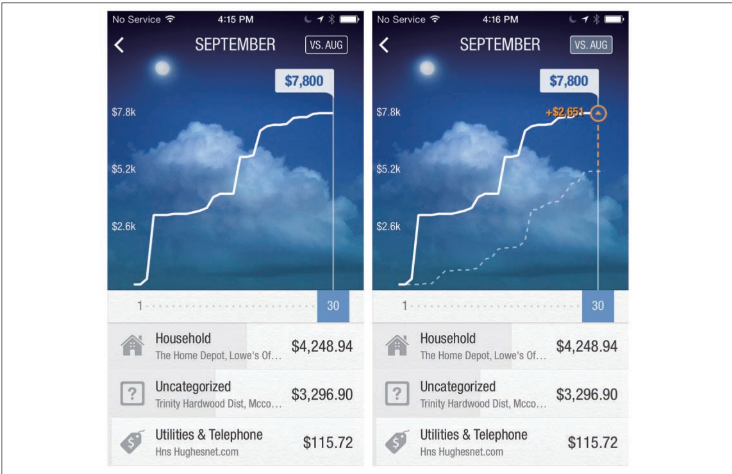



图 3-10

iOS 版 BillGuard：交互式概览图表

注 1：对比 8 月数据。（译者注）

[小贴士]

将概览放在数据行区块上方。如果图表概览太大，可以考虑用逐级深入模式代替。若想了解有效的图表设计技巧，可以参见第6章。

如果概览用饼图或环形图表示，且图表和图例占用了整个页面，那么采用逐级深入（Drill Down）模式的效果可能会更好，就像Roambi设计的那样。更多示例请参见第6章。

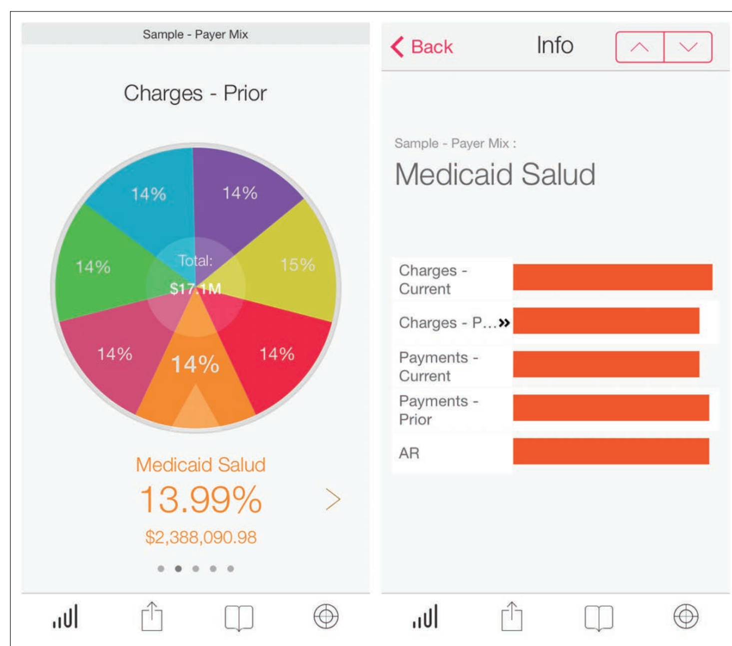



图3-11

iOS版Roambi：图表占用了整个页面，所以逐级深入模式比概览加数据模式效果更好

## 行分组表格

行分组可以让表格数据更易读。一组分组行集可以作为节区头部，比如按年分组（例如Zillow Mortgage Calculator）或按类别分组（例如Mint）的交易记录。

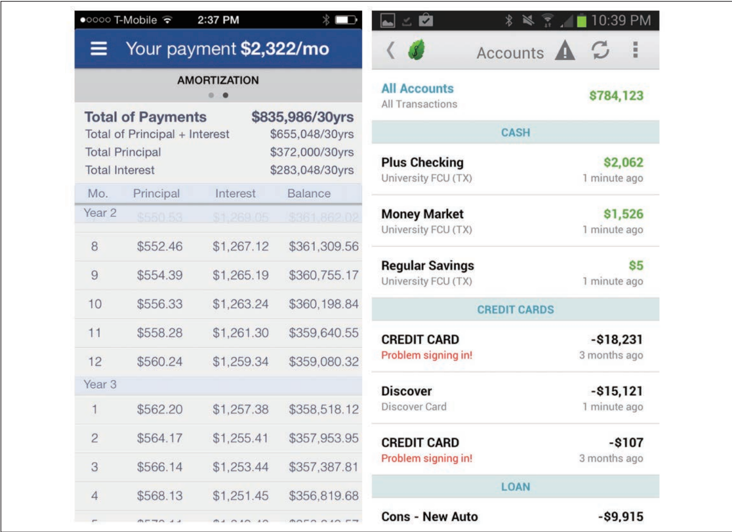



图 3-12
安卓版 Zillow Mortgage Calculator 和 Mint：行分组表格

［小贴士］

从视觉上将表格的汇总行和其他行区别开来。一般而言，汇总行应该较窄，字体颜色较浅。

## 带有视觉指示器的表格

带有视觉指示器的表格模式设计中，波形图（Sparklines）和图标可以提升表格的信息显示效果。但一定要注意，不要用没用的图标堆满表格，给数据的显示造成干扰，就像 Sleep Charts 的设计那样。




图 3-13
安卓版 Sleep Charts：不要在表格里填满不必要的图标

另一方面，Roambi Sales Reports 应用的视觉设计突出了数据，而不是使之模糊不清。关于在移动应用中使用波形图的更多信息，请参见第 6 章。

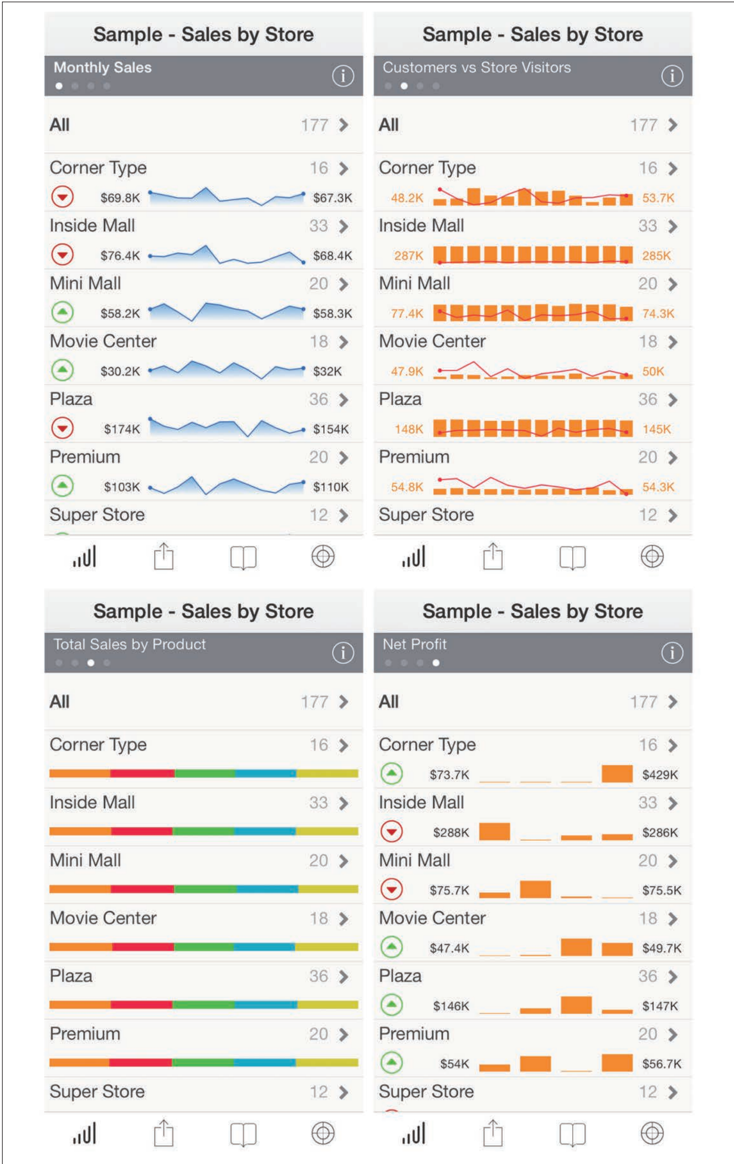



图 3-14

iOS 版 Roambi Sales Reports：包括波形图在内的设计元素突出了数据信息

要选择可以快速被识别的视觉指示信息。再强调一次，避免使用不必要的图标。比较一下最近的两个 iOS 版本的 PayPal。较新版的 iOS 7 设计去掉了过多的图标，增加了更多的空间来展示交易信息。

［小贴士］

只有当视觉指示器可以突出表格数据时，才可以使用。否则，应该避免使用。

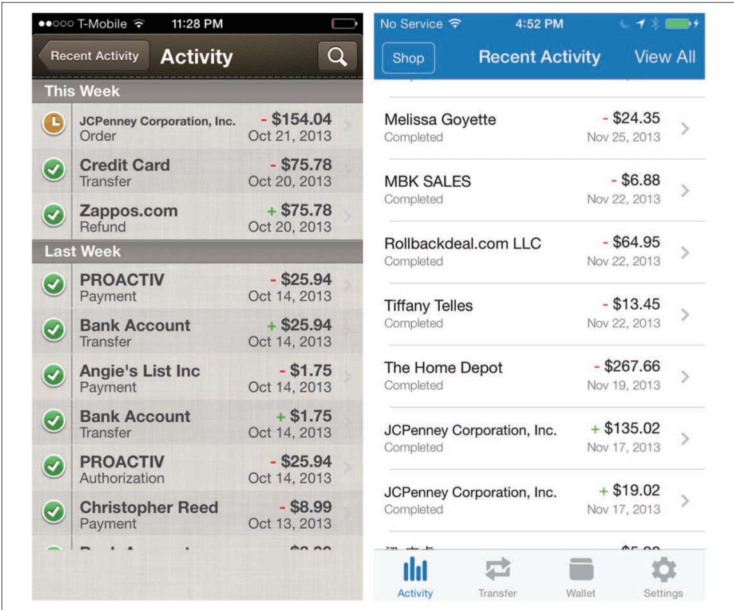



图 3-15

iOS 6 版和 iOS 7 版 PayPal：新版的设计丢弃了图标，增强了可读性

## 可编辑表格

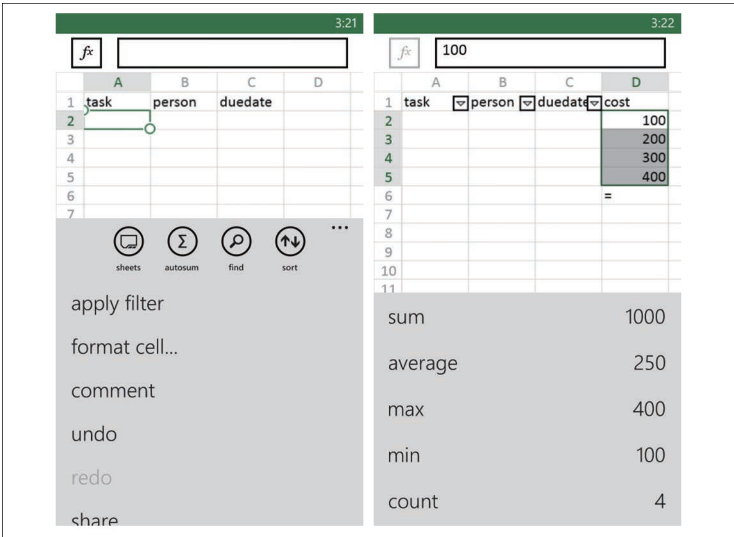
在移动界面设计中，可编辑表格几乎为微软 Office、谷歌云端硬盘（Google Drive）和苹果 Numbers 等电子表格应用所独有。许多网页端可编辑表格的设计规范也适用于移动端：

- 清晰地指示出选中的单元格和行。
- 如果单元格有特定的格式，就要提供适当的编辑器（如选择器、下拉菜单、颜色选择器、日期选择器等）。
- 除非编辑后产生错误提示，否则没有必要为每项编辑都提供确认反馈，只需在保存时要求用户确认即可。

在 Windows Phone 版的 Office 中，单击单元格会显示单元格菜单选项。点击功能图标（左上角的 fx 按钮），以调出功能菜单。

图 3-16

Windows Phone 版 Office：点击 fx 按钮调出功能菜单




［小贴士］

设计可编辑表格时，可适当地参照网页端和桌面端设计的最佳实践，但是如果要进行批量数据输入，或者可能需要大量的编辑工作，就要避免使用可编辑表格。

然而，与桌面应用不同的是，移动端的可编辑表格不适合进行大量的数据输入操作，这主要是因为移动设备的键盘不支持屏幕导航，尤其是不能使用 Tab 键在单元格之间切换。

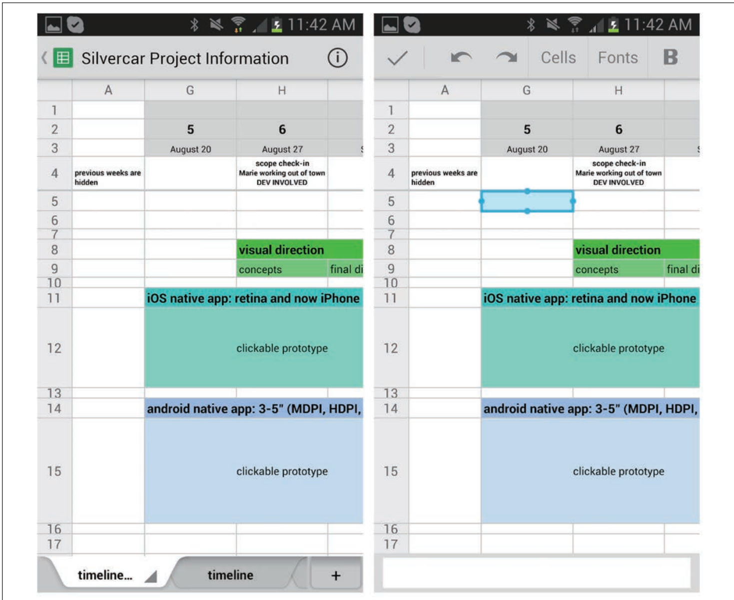



图 3-17

安卓版谷歌云端硬盘：选择一个单元格时，屏幕底端会显示输入框

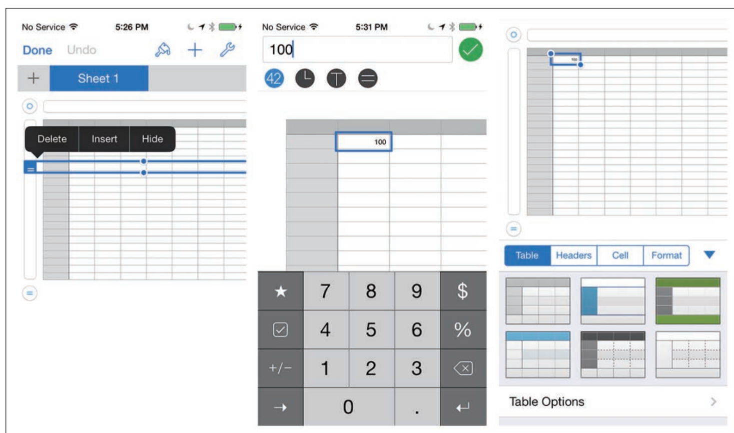



图 3-18

iOS 版 Numbers：对数据的输入和格式化分别采用不同的视图和编辑模式

# 第 4 章

# 搜索、排序和筛选




- **搜索模式**

  隐式搜索（Implicit Search），显式搜索（Explicit Search），自动完成搜索（Auto-Complete），限定范围的搜索（Scoped Search），动态搜索（Dynamic Search），保存的搜索（Saved）、最近搜索记录（Recent）和热门搜索（Popular），搜索表单（Search Form），搜索结果（Search Results）

- **排序模式**

  单页排序（Onscreen Sort）、排序浮层（Sort Overlay）、排序表单（Sort Form）

- **筛选模式**

  单页筛选（Onscreen Filter）、筛选浮层（Filter Overlay）、筛选表单（Filter Form）、筛选抽屉（Filter Drawer）、基于手势的筛选（Gesture-

Based Filter）

你最近看过 AutoDirect、Greensheet 或类似的免费广告册吗？你会看到一页页的广告分放在众多不同的类别下面，比如卡车、小汽车和电器。要想查看有没有感兴趣的广告，你得把册子从头到尾翻一遍。这应该会使你对将要讲到的三种基本在线工具产生全新的认识——搜索、排序和筛选。

在数字领域，我们将搜索、排序和筛选视为理所当然的工具。但是要将这三者在移动应用中很好地实现，还是稍有差别。本章将探讨十几种不同的实现方法。

## 搜索模式

《触动人心：设计优秀的 iPhone 应用》（*Tapworthy: Designing Great iPhone Apps*）一书的作者，Josh Clark 说过，在触屏设备的界面设计中，“按钮只是一个变通手段”。这是因为，触摸让我们能更直接地与内容进行交互，比如下拉刷新和双指缩放。那么，在使用肢体动作更直观的情况下，为什么要设计一个抽象的按钮呢？

我们应该以同样的方式去思考移动搜索。随着我们的应用越来越能“感知”个体的需求，它们也在减少我们必须运用显式搜索查找相关信息的情况。

比如说，谷歌“知道”我要在市区参加一个会议，它会告诉我应该在什么时间出发，如何到达。它还知道，我经常在周六带孩子们去看电影，所以它“推荐”了一些正在我们最喜欢的剧院上映的演出。当我在有我喜欢的几家店铺的商城附近时，RetailMeNot 会为我显示出这些店铺此时的所有打折信息。

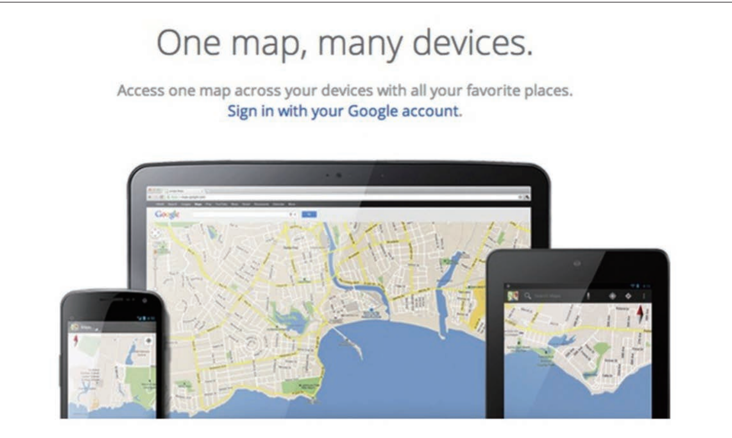
这种新兴的模式被称为隐式搜索。我们会进一步探索这种模式，重新审视本书上一版中谈到的传统搜索、排序和筛选技术。

在这一节中，我们将讨论针对移动应用的搜索模式，其中包括隐式搜索，显式搜索，自动完成搜索，限定范围的搜索，动态搜索，保存的搜索、最近搜索记录和热门搜索，搜索表单以及搜索结果。

## 隐式搜索

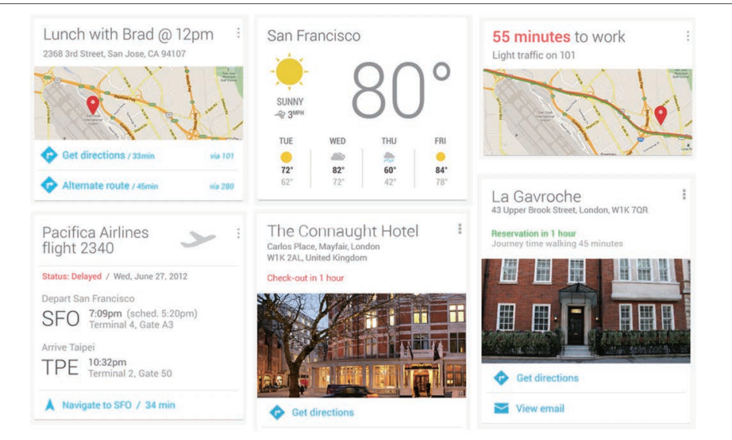
在与一位新客户会面的前一天晚上，我在电脑上通过谷歌搜索了这位客户的地址。第二天早上临走前，我打开新版谷歌地图，输入地址的第一个字母，地图马上就知道我要去哪儿。

图 4-1
谷歌地图：跨平台整合




在这个新版应用中，谷歌跨平台同步了我的信息，并用隐式搜索的方式显示出来。“哇，太有用了。”我当时想。但在那之后，Google Now 又把隐式搜索提升到了一个新的层次，它在地图上标出了我的下一个与会地点、应该出发的时间，以及到达的途径。我现在已经基本上不需要在谷歌地图里进行精确的搜索了。

图 4-2
Google Now：把隐式搜索提升到了新的层次



在使用 RetailMeNot 应用时，我也经历了相似的愉快体验：我开车外出办事的时候，突然听到包里发出了“咔哧”一声响。我很纳闷，我的手机怎么会发出收银机似的声音，这时我看到了一条 RetailMeNot 的提醒——附近商城里我最喜欢的几家店铺正在进行大力度促销。当我走进一家商铺时，RetailMeNot 应用又会显示出能在购物时使用的优惠券。要知道，是这条与我相关的信息找到了我，而不需要我去搜索它。

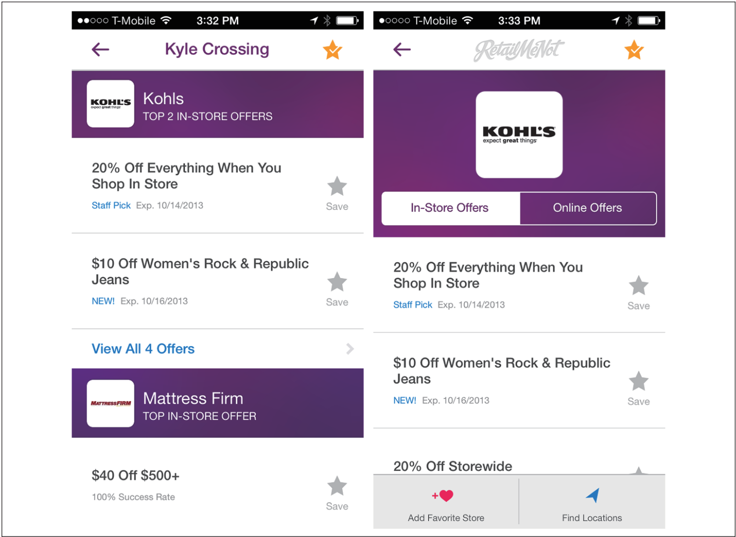



图 4-3

iOS 版 RetailMeNot：基于附近位置的隐式搜索

对我来说，这种方式比先通过精确搜索找到一家商店，然后再查看该商店是否提供任何打折信息要高效得多。这对 RetailMeNot 公司来说当然也是件好事，自从推出地理围栏（Geofencing）[1] 功能后，其移动应用的活跃量增长了四倍之多。

隐式搜索的例子还有很多。比如，如果在 Zillow 和其他类似的房产应用中开启了基于地理位置的服务，应用就会立即显示所在区域内的房产销售和租赁的全部信息。交友应用 Badoo 使用了同样的技术来显示附近正在找朋友的人。

[小贴士]

在选择其他搜索模式前先考虑隐式搜索的可能性。这种模式可以创造愉悦的用户体验。

注 1：地理围栏是 LBS 的一种新应用，就是用一个虚拟的栅栏围出一个虚拟地理边界。当手机进入、离开某个特定地理区域，或在该区域内活动时，它可以自动接收通知。（译者注）

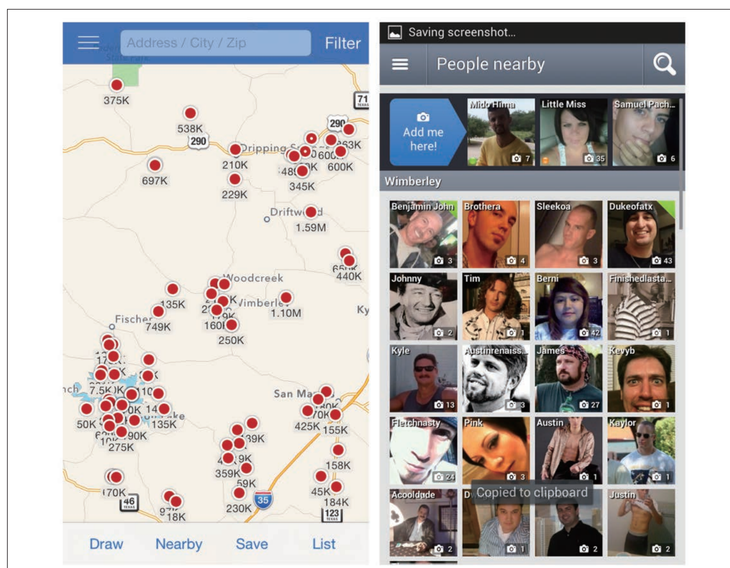



图 4-4

iOS 版 Zillow 和安卓版 Badoo：基于地理位置的隐式搜索

Foodspotting 和 Hungry Now 在它们各自的着陆页上突出了“查看附近的餐厅”这一功能。Foodspotting 使用轮播图显示每个餐厅的图片和详情，鼓励用户根据自己的就餐体验为餐厅打分，而 Hungry Now 则在一张地图上显示出附近的餐厅。

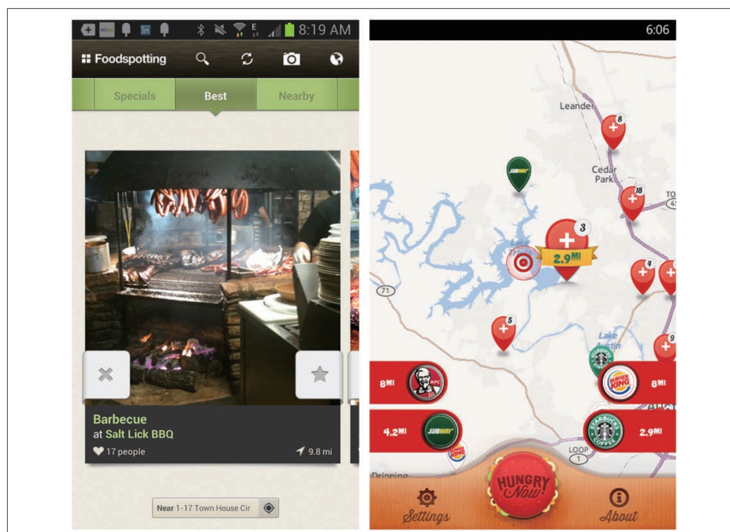



图 4-5

安卓版 Foodspotting 和 Windows Phone 版 Hungry Now：更多基于地理位置的搜索结果

像 Wikitude 和 Layar 这样的应用结合使用了移动设备的地理位置和加速计，为用户实现了“增强现实”（Augmented Reality）功能，且无需输入搜索词。

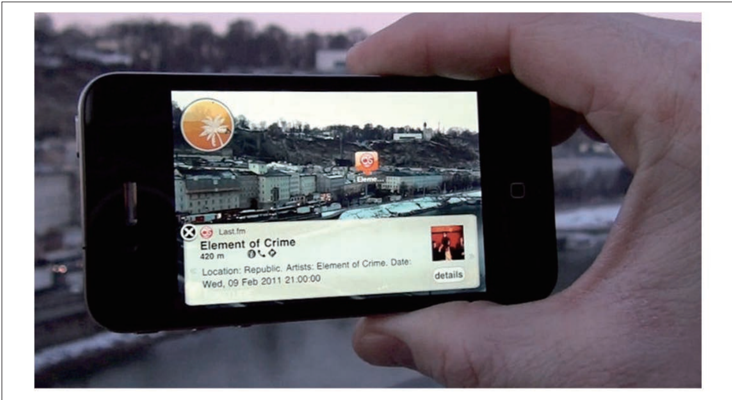



图 4-6
Wikitude World Browser：使用增强现实的隐式搜索

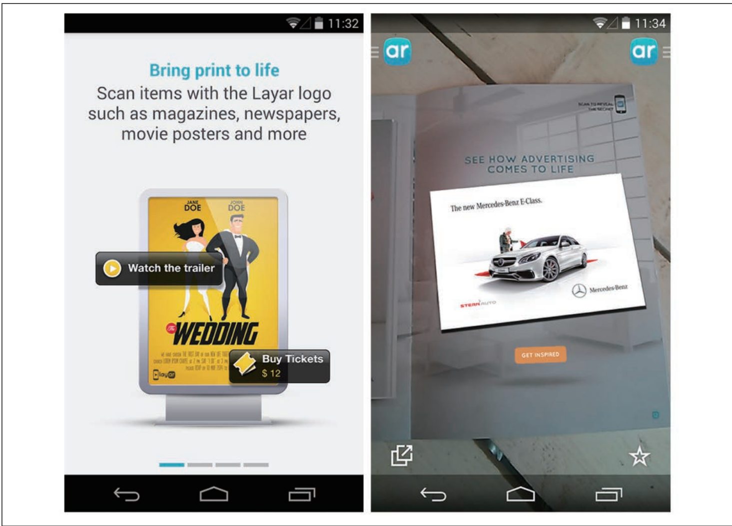


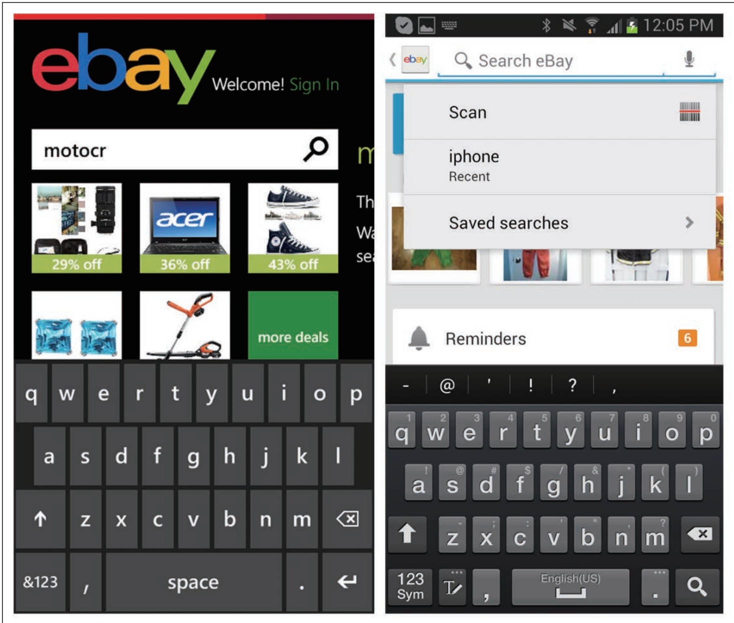

图 4-7
安卓版 Layar：另一种使用增强现实实现隐式搜索的方法

## 显式搜索

虽然隐式搜索可以创造出更丰富的用户体验，但显式搜索模式仍然有其用武之地。顾名思义，显式搜索就是通过显式的用户操作实现搜索并获得结果。

Windows Phone 8 版的 eBay 应用就是个简单的例子。在搜索框内输入检索词，然后点击搜索图标，或者敲击键盘上的回车键查看搜索结果。但是安卓版的 eBay 只是增加了几个小功能，就比 Windows Phone 8 版有用得多。

图 4-8
Windows Phone 版和安卓版 eBay




在安卓应用中，点击搜索框会出现几个选项：扫描商品而不用输入商品名，再次使用最近一次搜索，或是访问保存的搜索。只要用户开始输入，应用就会给出与输入文字匹配的搜索建议。以这种方式最小化输入需求，可以提高搜索效率，并且降低出错的几率。关于这一点，也可参见接下来的模式——自动完成搜索。

正如安卓版 eBay 的搜索设计那样，显式搜索不应该仅限于文字输入。亚马逊应用和 RedLaser 应用就为快速查找店内商品提供了图片和条形码扫描的功能。这些对移动设备友好的搜索功能促进了展厅现象（Showrooming Phenomenon，http://en.wikipedia.org/wiki/Showrooming）[1] 的发展，该现象在近几年已改变了零售业的发展势态。

注 1：展厅现象指先在商场看好商品进行体验，然后上网搜索并以低价购买的现象。（译者注）

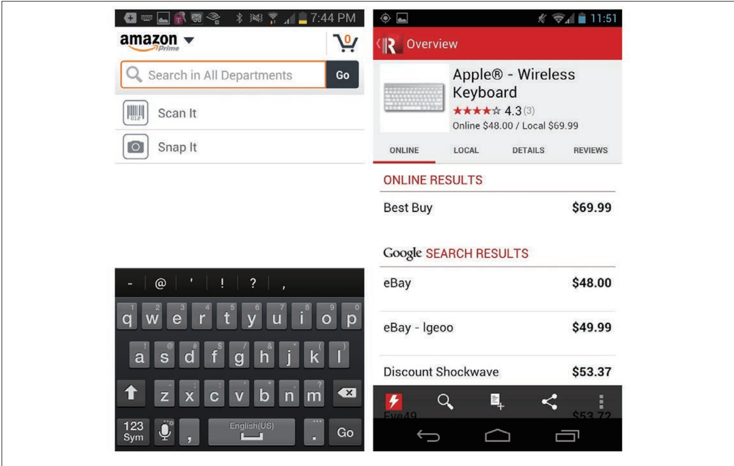



图 4-9

安卓版亚马逊和 RedLaser：通过扫描条形码进行显式搜索

语音搜索也同样有用，但依赖于高质量的语音识别程序。我们都听过 Siri 严重误解用户问题的搞笑故事（请看来自 WhySiriWhy.com 网站的截图），但语音搜索已经取得了很大进展，我期待它能够得到持续改进。

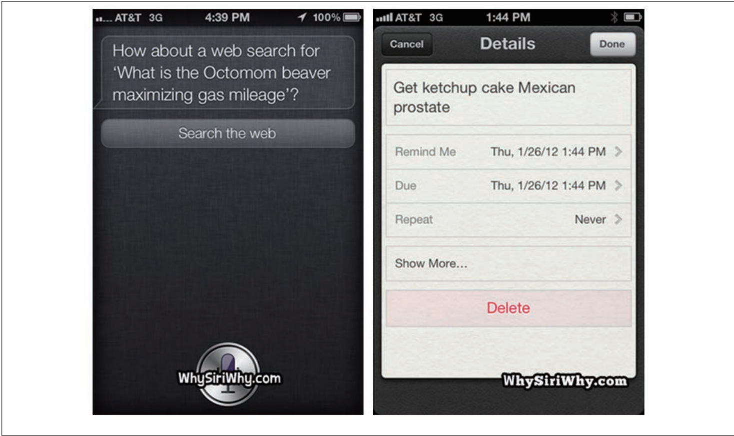



图 4-10

WhySiriWhy.com：一个专门发现语音搜索错误的网站

还有一种显式搜索是通过直接操纵（Direct Manipulation）[1] 完成的。Trulia 和 Soundwave 都提供了用手势在地图上划出搜索范围的功能。

注 1：直接操纵指用光笔、鼠标、触摸屏或数据手套等坐标指点设备，直接从屏幕上获取形象化命令与数据的过程。（译者注）

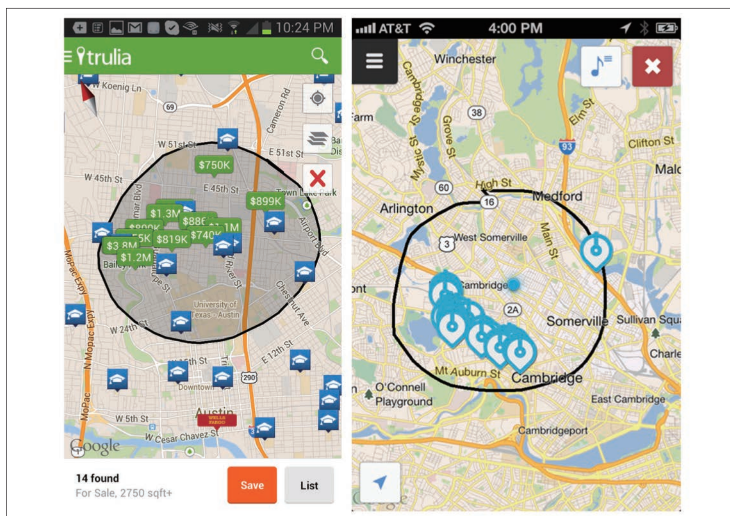



图 4-11
安卓版 Trulia 和 iOS 版 Soundwave：使用手势划出搜索边界

还有一种新兴的显式搜索模式是“模态搜索”（Modal Search）。以 Target 为例，其搜索框位于导航抽屉内，但是只要点击搜索框，“搜索模式”就会显示为一个带有弹出键盘的完整的搜索页。

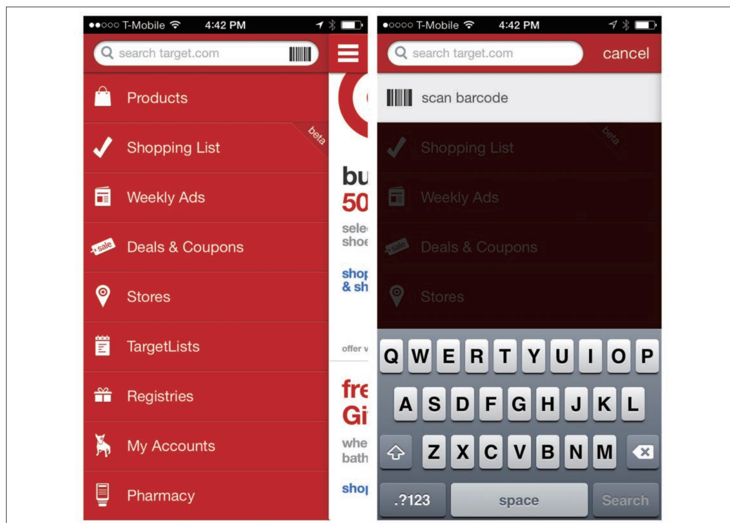



图 4-12
iOS 版 Target：点击搜索框打开一个新的搜索页面

TripAdvisor 和 RetailMeNot 使用了同样的搜索模式，但它们的搜索框位于首页，而没有放在导航抽屉里。

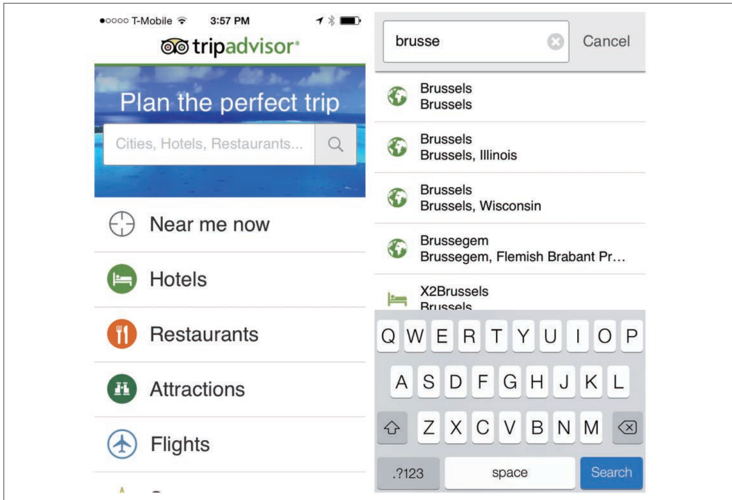



图 4-13
iOS 版 TripAdvisor：模态搜索页面从右向左滑出

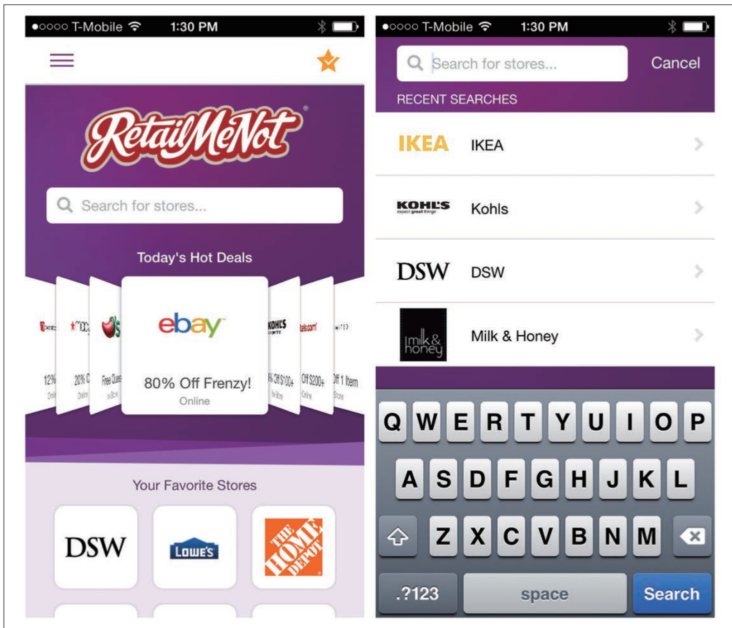



图 4-14
iOS 版 RetailMeNot：模态搜索页面从下向上滑出

模态搜索的主要好处在于用单个搜索页面增加了搜索空间。记住，为那些最终不打算搜索的用户提供一个返回的途径（比如“取消”按钮）。

[小贴士]

要提供一个“清除”按钮和取消模态搜索的选项。使用反馈来表示系统正在进行搜索操作（参见第 8 章）。

## 自动完成搜索

现在移动设计已进入成熟阶段，自动完成功能应该成为显式搜索模式的标配。虽然这些设计模式从 2004 年左右开始就已经出现在网页端，桌面端还要更久一些，但在我研究过的 Windows Phone 8 平台的应用中，只有不到 10% 的应用拥有这项功能，这着实让我吃了一惊。更吓人的是，甚至连 Windows Phone 版的 Dictionary.com 都没有提供自动完成功能，还有谁比词典应用的用户更需要这项功能呢？

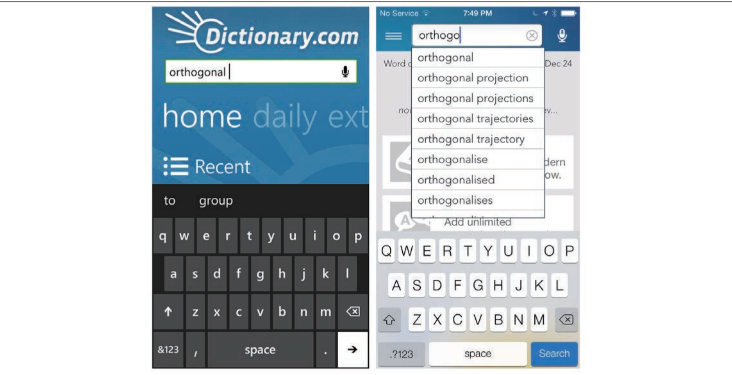



图 4-15
Windows Phone 版和 iOS 版 Dictionary.com：自动完成应该成为显式搜索模式的标配

应该配置自动完成功能，这样当用户输入文字时，就会出现一组可能的结果。当用户继续输入文字时，搜索结果也相应地减少。点击任何一项结果就会触发搜索。点击页面上或键盘上的按钮，也应该触发对当前高亮选项的搜索。

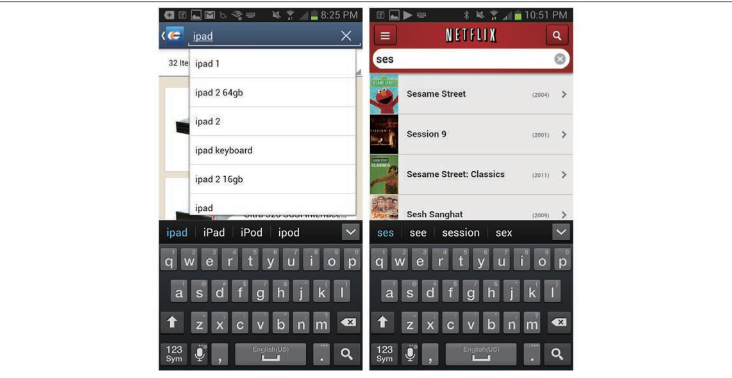



图 4-16
安卓版 Newegg 和 Netflix：自动完成功能可以为触发搜索提供多种方式

理想情况下，自动完成搜索的结果会立即显示出来，但应当使用进程指示器（“搜索中……”）给予用户反馈。Fidelity 应用就在最终搜索结果页面上显示反馈。

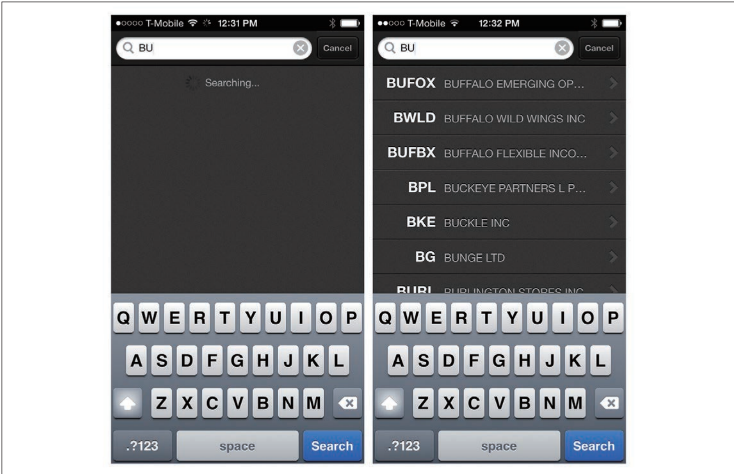



图 4-17
iOS 版 Fidelity 在搜索框下方提供自动完成的进程反馈

安卓版 Target 应用设计了一种增强型的自动完成搜索，它将每一项提示有可能所属的类别都显示出来。这样的设计令用户可以更方便地直接进入最相关的搜索结果。Songza 也采用了显示群组名的增强型自动完成搜索。

［小贴士］

如果在显示自动完成搜索的结果前存在延迟，那么可以显示一个加载指示器。可以考虑将提示进行分组，以提高搜索效率。

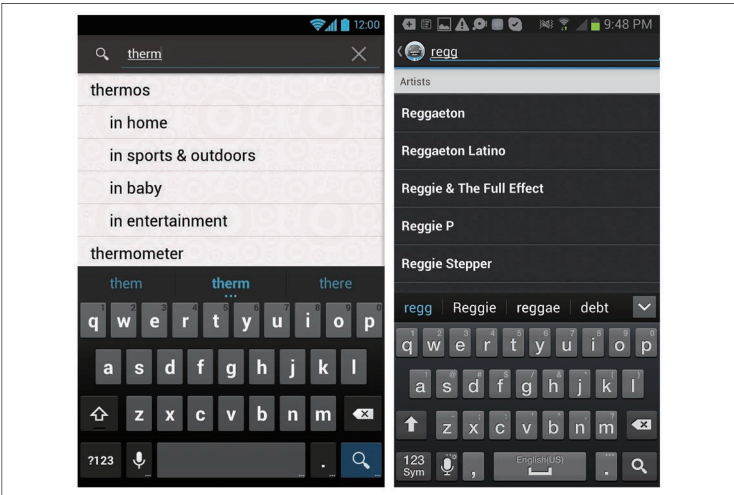



图 4-18
安卓版 Target 和 Songza：增强型自动完成搜索

## 动态搜索

你也可以把动态搜索模式看成动态筛选。在动态搜索模式下，空白的搜索框下会显示全部可能的分类结果，而输入搜索字符串会动态地对结果进行筛选，缩减列表的范围。以 Audible 为例，你可以看到我的整个有声书库显示在背景层。当我开始输入的时候，只有与输入名称匹配的书籍还显示在页面上。

图 4-19

iOS 版 Audible 的动态搜索在用户输入时缩减搜索结果的范围

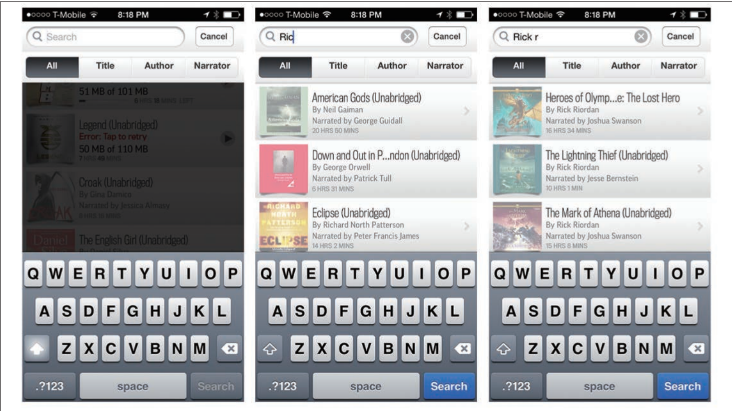
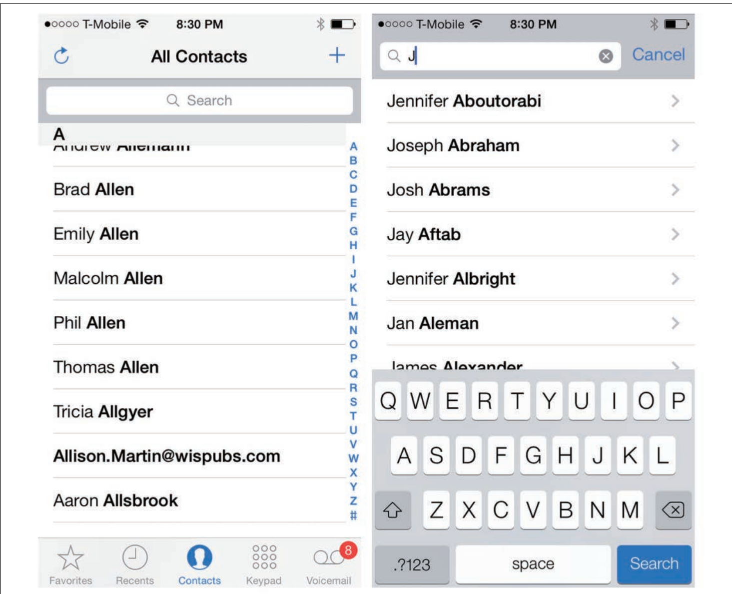



图 4-20

iOS 版 Contacts 中的动态搜索允许用户快速对搜索结果进行筛选



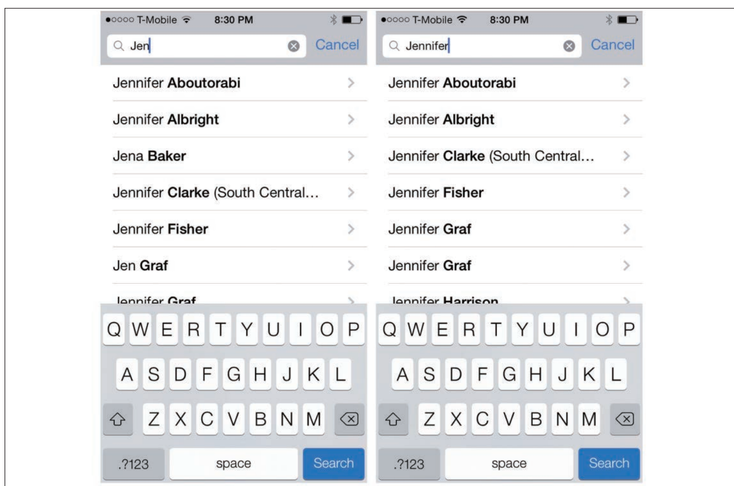



图 4-20（续）
iOS 版 Contacts 中的动态搜索允许用户快速对搜索结果进行筛选

［小贴士］

动态搜索非常适用于有限的数据集，比如地址簿或是个人媒体库，但在搜索零售商存货清单这样的海量数据时，则不应该采用这种模式。

## 限定范围的搜索

有些时候，如果让用户在开始搜索前先设定好搜索范围，他们就能更便捷地得到搜索结果。Windows Phone 8 版的谷歌应用提供了一个可以选择搜索类别的菜单，而安卓版的 Google Play Store 则在搜索框下面放置了一个下拉菜单控件。

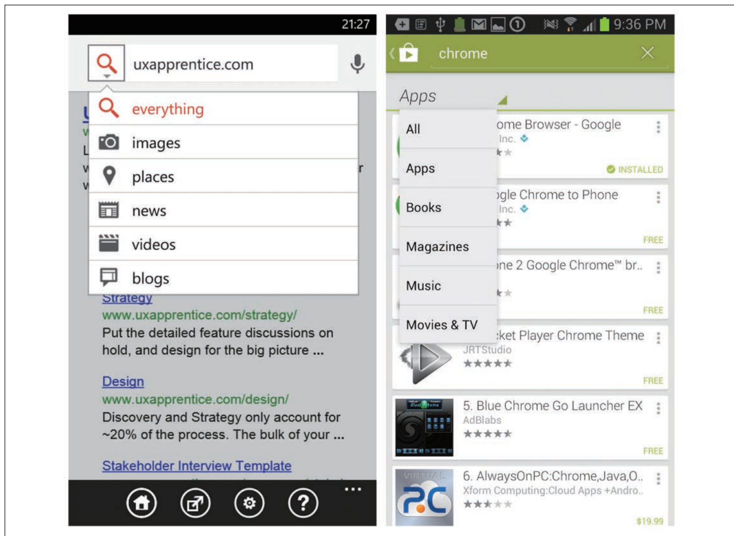



图 4-21
Windows Phone 版的谷歌应用和安卓版的 Google Play Store：用菜单选项限定搜索范围

［小贴士］

为数据集提供合理的范围选项，三到五个就够了。如果要提供高级搜索，可以考虑设计成一个搜索表单。

在 iOS 平台上，常常用分段控件（Segmented Control）实现限定范围的搜索，比如 iTunes，但也可以选用其他的 UI 控件，比如滚动选项卡或胶囊导航栏（Pill Bar）。

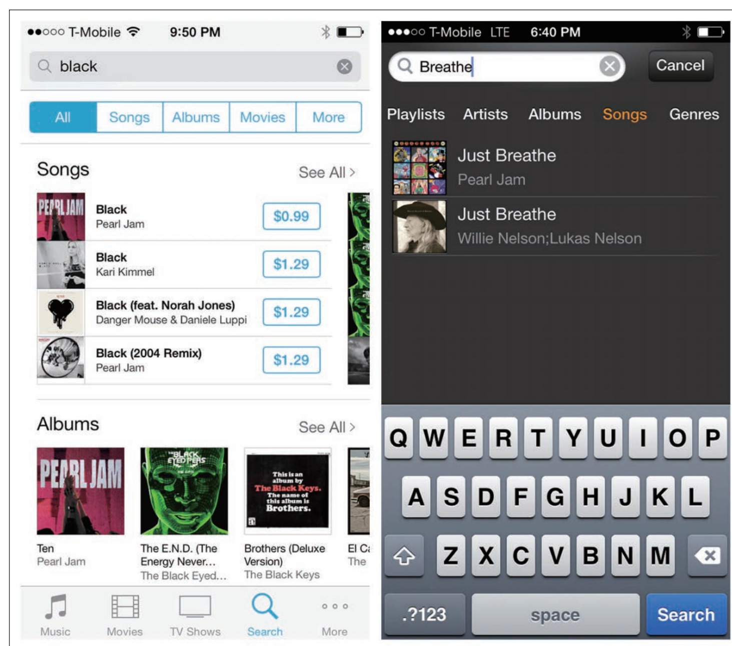



图 4-22

iOS 版 iTunes 和 Amazon Cloud Player：两种缩小搜索范围的设计

## 保存的搜索、最近搜索记录和热门搜索

优秀的移动应用界面设计都会遵循基本的可用性准则：尊重用户付出的努力。保存的搜索、最近搜索记录和热门搜索就是通过选择先前的搜索记录，实现搜索流程的简化，让用户不必重复输入相同的关键词或搜索条件。

［小贴士］

为便于日后使用，通常需要增加一个步骤来为保存的搜索命名，而最近搜索记录是自动保存并自动显示的。可以想想哪种模式能更好地满足用户的需求。

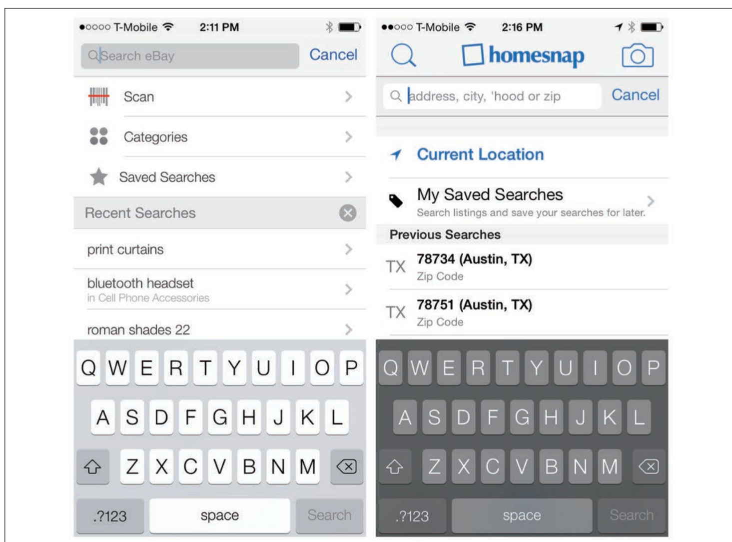



图 4-23

iOS 版 eBay 和 Homesnap：保存的搜索和最近搜索记录

Groupon 直接显示出用户的热门搜索，CheckPay 则把热门搜索作为搜索的备选项。

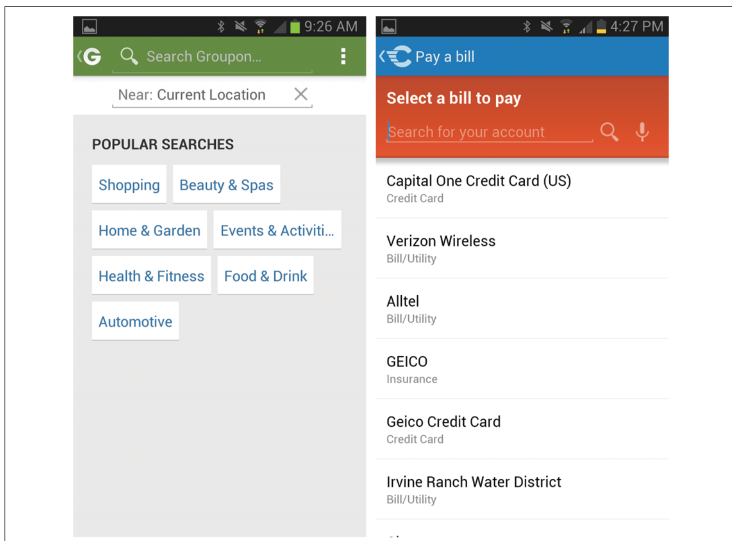
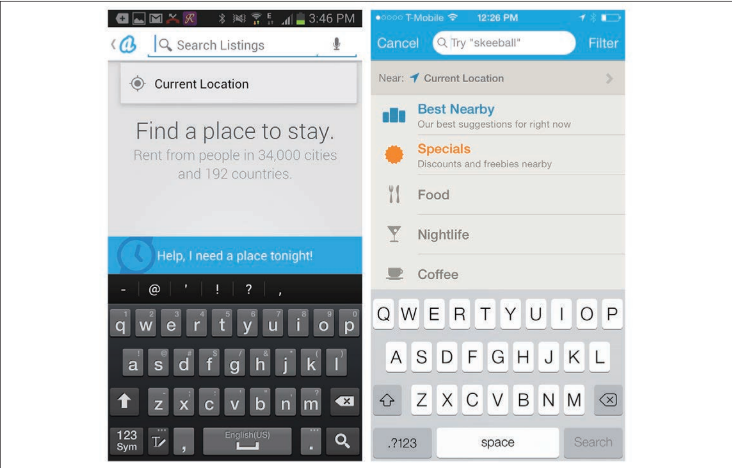



图 4-24

安卓版 Groupon 和 CheckPay：显示热门搜索

还有其他一些尊重用户努力的技巧，其中包括 Airbnb 和 Foursquare 采用的基于地理位置的搜索。

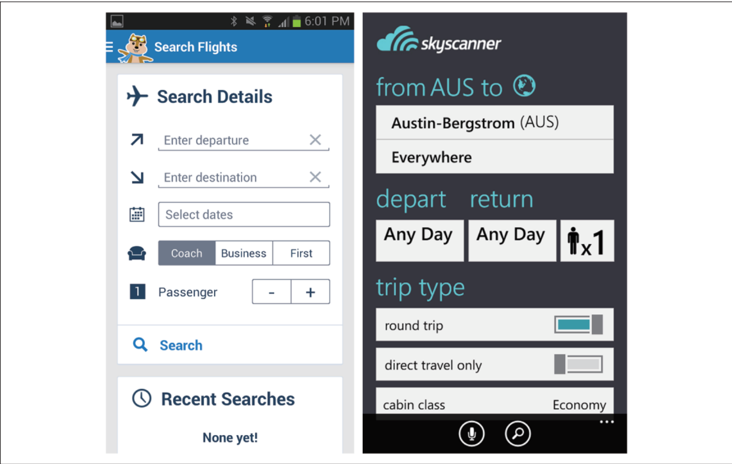
图 4-25
安卓版 Airbnb 和 iOS 版 Foursquare：基于地理位置的搜索



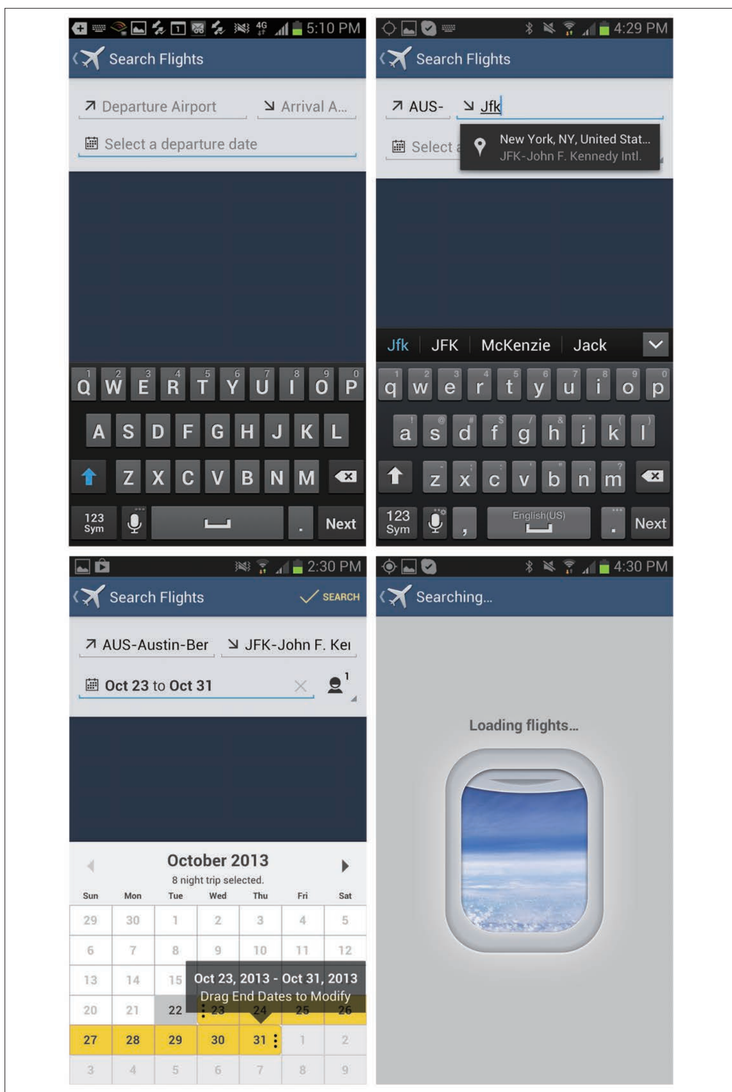

## 搜索表单

这个模式的特点是，有一个用于输入多个搜索条件的单独的表单和一个显式搜索按钮。Hipmunk 和 Skyscanner 的机票和酒店搜索就采用了搜索表单模式进行必要条件的设置。更多示例，请参见第 2 章。

图 4-26
安卓版 Hipmunk 和 Windows Phone 版 Skyscanner：搜索表单示例




iOS 版和安卓版的 Expedia 都在起点和终点字段上使用了美观的动画效果。这两个版本也都只设计了必不可少的搜索字段。




［小贴士］

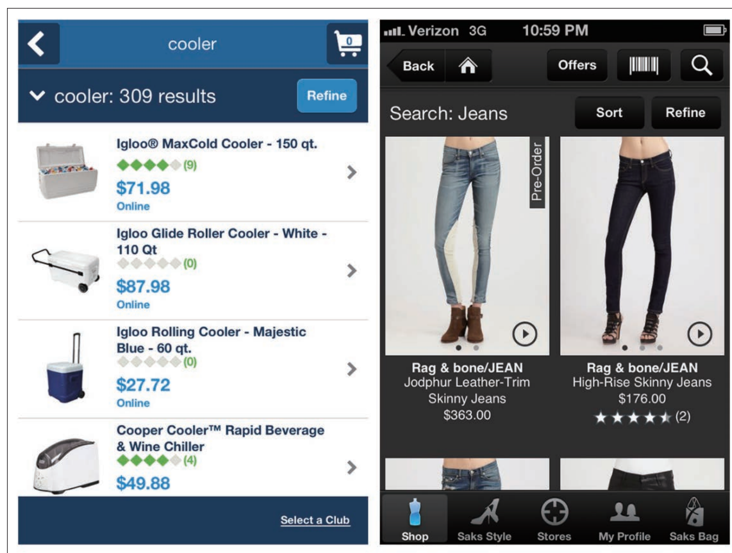
最小化输入字段的个数。正确使用操作系统特定的输入控件。仿照表格设计的最佳范例进行对齐、标签、字号等方面的设计。

图 4-27
安卓版 Expedia：实用又美观的搜索表单设计

## 搜索结果/查看结果

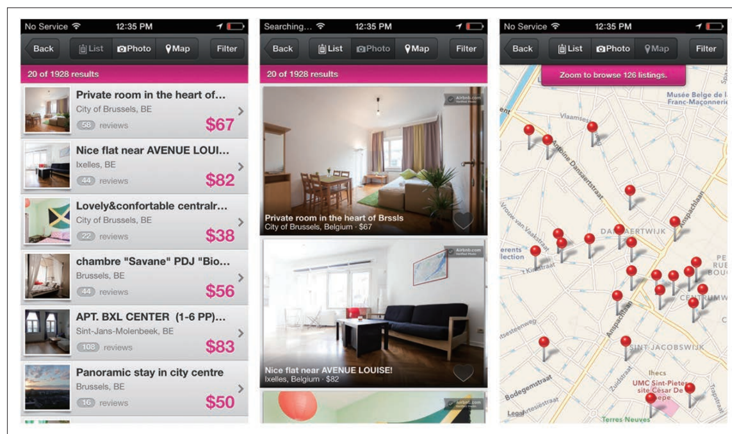
在执行搜索后，搜索结果可以在同一页面显示，也可以跳转到专门的搜索结果页。搜索结果可以显示为表格、列表、地图、卫星图或缩略图的形式。

图 4-28
iOS 版 Sam's Club 和 Saks：搜索结果分别采用了列表视图和磁贴视图




根据搜索结果的类型和用户偏好，也可以提供多种视图选项。

图 4-29
iOS 版 Airbnb：列表、图片和地图形式的搜索结果视图




还有一种展示搜索结果的新趋势是，用卡片式、幻灯片或磁贴的样式将搜索结果一一呈现出来。用户可以用轻扫的手势进行切换浏览。

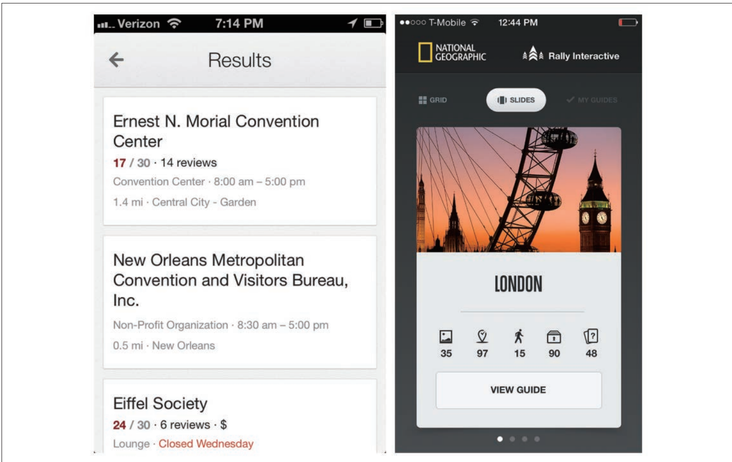



图 4-30

Google Now 的卡片式搜索结果及 National Geographic 的幻灯片式搜索结果

Hipmunk 采用甘特图（Gantt chart）式时间线设计，开创性地对机票的搜索结果实现了可视化。Expedia 则视觉化了飞行时长，同时显示出机票价格和起飞时间。

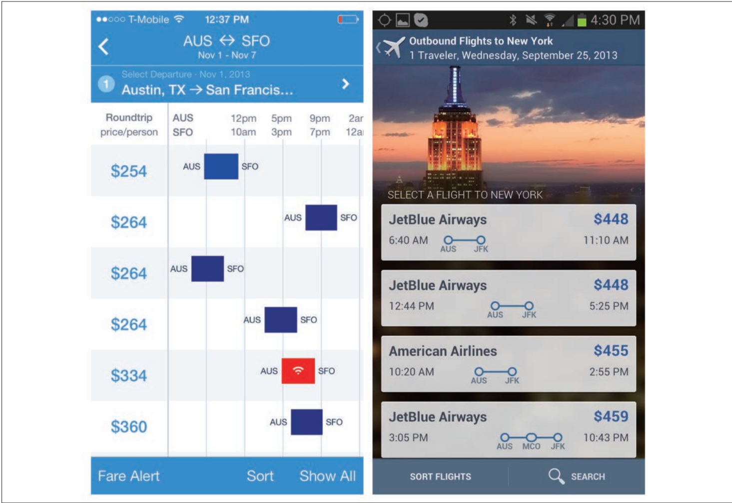


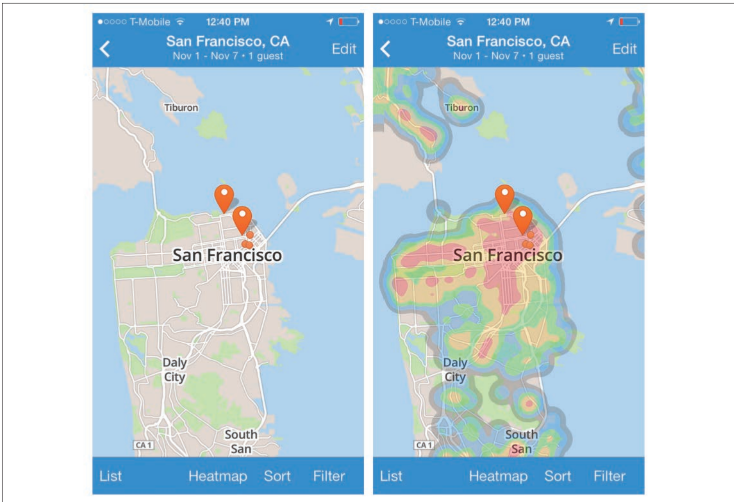

图 4-31

iOS 版 Hipmunk 和安卓版 Expedia

Hipmunk 还提供了查看酒店搜索结果的另一种视图。将搜索结果显示在地图中——很标准，是不是？但还增加了一个热力图浮层的选项，用来查看酒店周围的犯罪区域，以及购物和夜生活方面的情况。

图 4-32

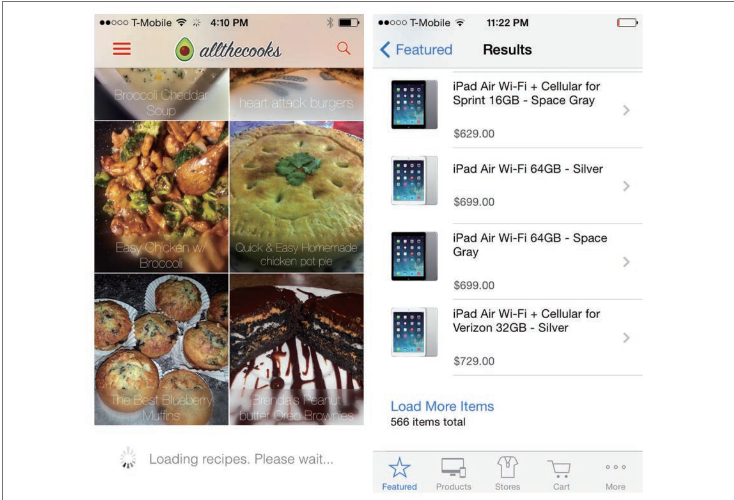
iOS 版 Hipmunk：用热力图浮层显示各种统计数据




延迟加载（Lazy Loading）是一种在其他结果正在加载时显示部分搜索结果的常用技术。Allthecooks 等许多应用都会在用户滑动屏幕时自动加载更多结果。Apple Store 应用则提供了一个链接来加载更多商品。

图 4-33

iOS 版 Allthecooks 和 Apple Store：延迟加载搜索结果




［小贴士］

标明搜索返回的条目个数。用延迟加载代替分页。设定合理的默认排序。

不要用分页表格显示搜索结果，这样做会破坏在移动设备上查看信息的自然交互模式。

## 排序模式

苹果因其“以用户为中心”的设计而受到广泛赞誉，但是我认为，App Store 的搜索功能不应该包括在内。比如，搜索“财务”，你会得到 2200 多个搜索结果。当你开始浏览这许许多多的应用时，可能会奇怪，这些应用是按什么顺序排列的呢？是按受欢迎程度、评分高低，还是发布日期？你无从知晓。很可能你要到别的地方去研究一下，看哪些财务应用最适合你，然后再到 App Store 专门搜索这些应用的名字。

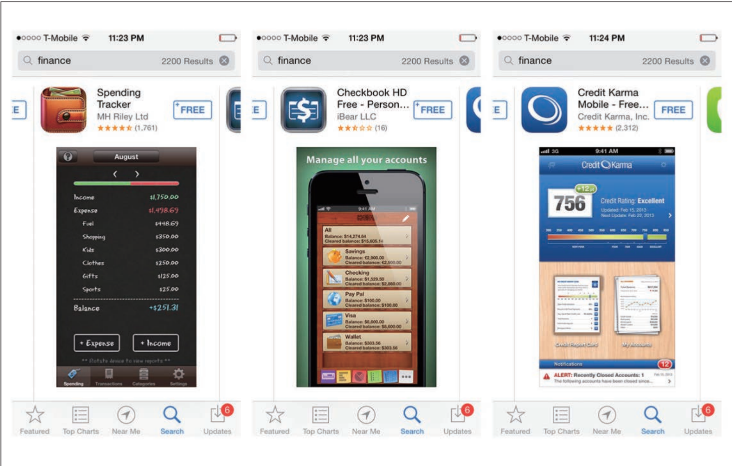



图 4-34

iOS 版 App Store：搜索财务应用返回的 2200 个未排序结果

同样，如果可以对谷歌地图的搜索结果进行操作就会方便得多。比如，我原以为“汽油”这个词的默认搜索结果会是离我所在位置最近的加油站，但我发现通常都不是这样。

实际上，我不得不打开搜索结果列表，查看每个搜索结果离我的距离。当然，这样还不够，因为距离和行驶时间不是一回事。如果可以将搜索结果按评价高低、距离远近或行驶时间进行排序的话，那就非常有意义了。唉，可惜没这个运气。

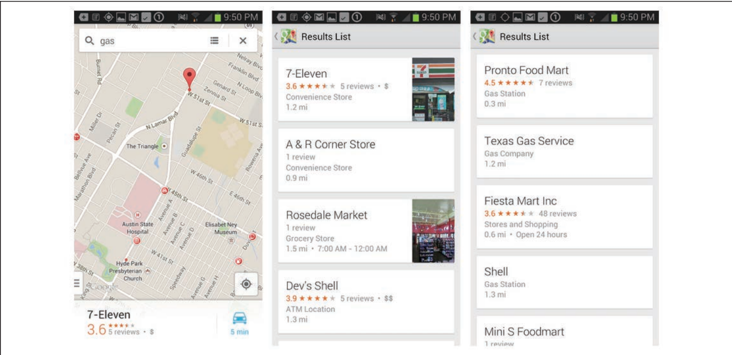



图 4-35
安卓版谷歌地图：在第二页上，距离最近和评价最高的搜索结果（Pronto Food Mart）被隐藏了

虽然在上面这些例子中，苹果和谷歌似乎忘了排序的价值，但排序模式确实能极大地提高搜索可用性。可以考虑在你的搜索结果中运用以下模式：

- 单页排序
- 排序浮层
- 排序表单

## 单页排序

在只需要少数几种排序选项的情况下，单页排序可以作为一种简单的一键式解决方案。但正如我们所见，许多 iOS 应用都弃用了这种模式，转而采用更节省空间的其他模式，比如排序浮层。

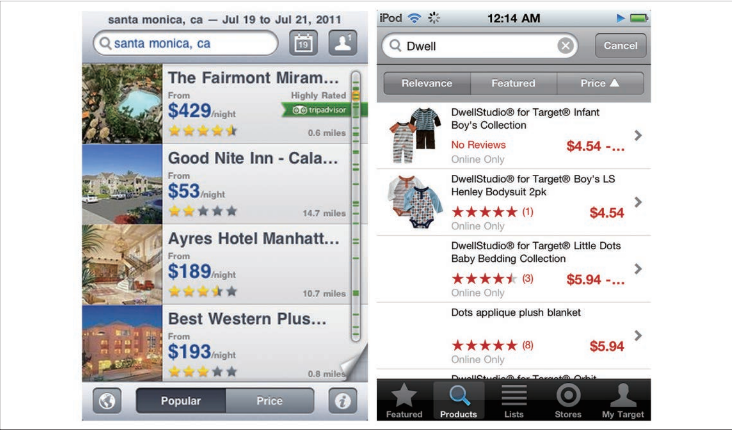



图 4-36
本书第 1 版中的单页排序示例：iOS 版 Expedia 和 Target

电子邮件应用 Boxer 只在点击排序按钮后才显示单页排序选项，用户点击关闭按钮后再隐藏。

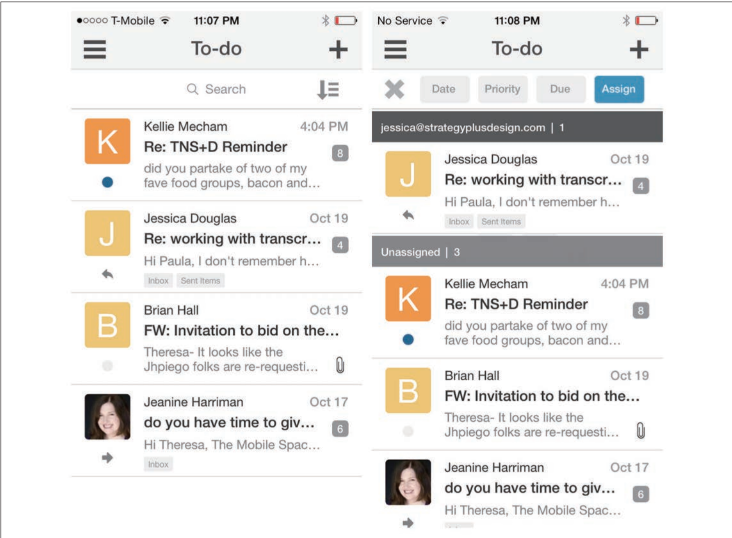



图 4-37

iOS 版 Boxer

在安卓应用中，下拉菜单控件可以很好地用来选择排列顺序。虽然不能一眼看到全部的排序选项，但它可以清晰地指明哪个选项是活动的。

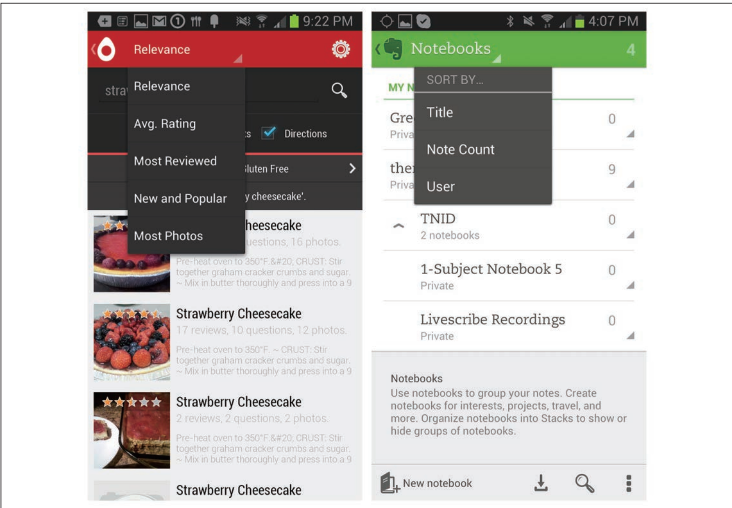



图 4-38

安卓版 Allthecooks 和 Evernote：下拉菜单显示排序选项

## 排序浮层

除了单页排序，排序浮层也是一种很好的备选方案。这种模式不会占用宝贵的屏幕空间，而且如果一个更长、更明确的标签效果更好的话，你也不必非用短标签不可。用户可以通过点击排序标签或图标开启浮层，可以将它放在搜索结果页面的顶部、中间或者底部。Audible 的排序设计是最为实用的一种，因为即便是在浮层关闭的时候，它也会把活动的排序选项显示出来。

图 4-39

iOS 版 Audible、Dropbox 和 eBay：位于顶端的排序浮层

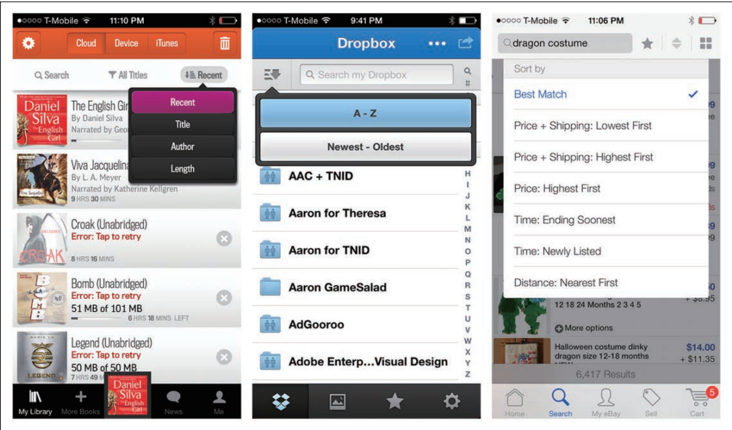
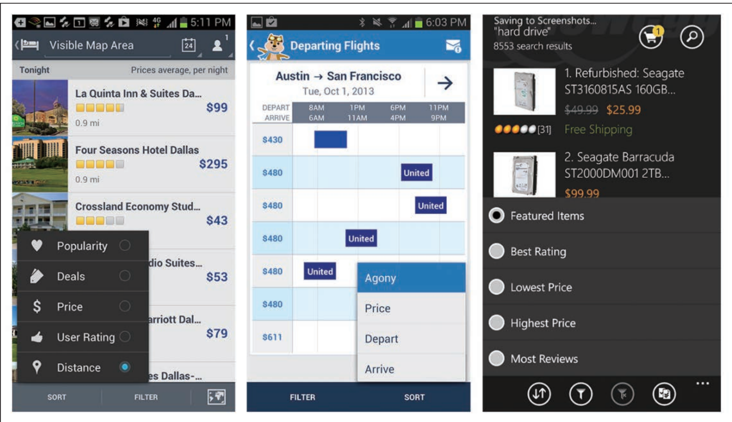



图 4-40

安卓版 Expedia 和 Hipmunk 以及 Windows Phone 版 Newegg：位于底端的排序浮层



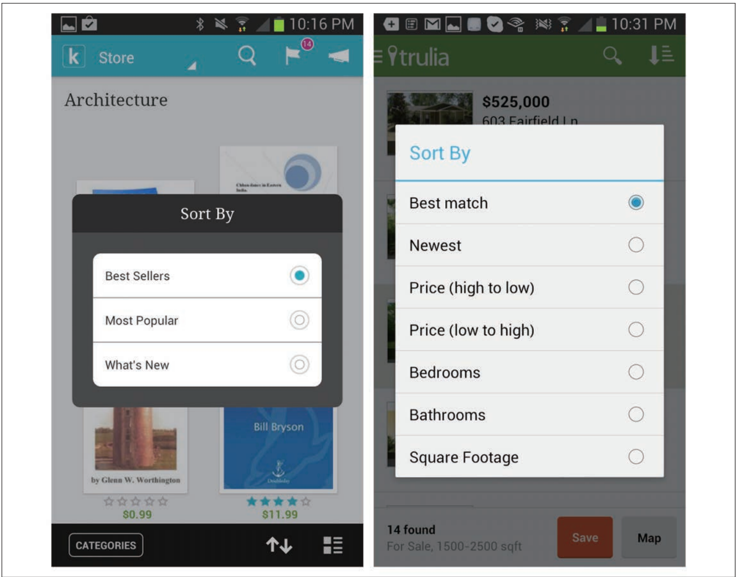



图 4-41
安卓版 Kobo 和 Trulia：对话框式的排序浮层

［小贴士］
排序选项应该放置于其他对结果列表进行处理的工具附近，比如筛选、视图切换开关，以及“清除搜索记录”按钮。要明确显示出活动的排序选项。

Wunderlist 提供了上下文工具，其中包括一个排序选项，可点击底端右下角进入。

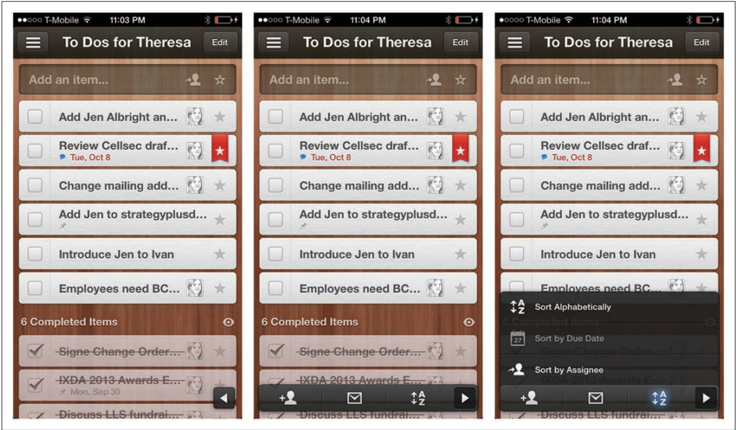
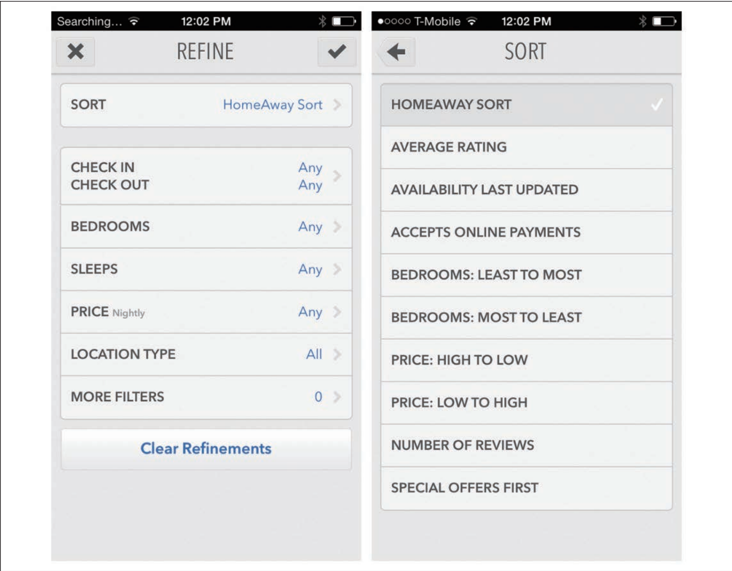



图 4-42
iOS 版 Wunderlist

## 排序表单

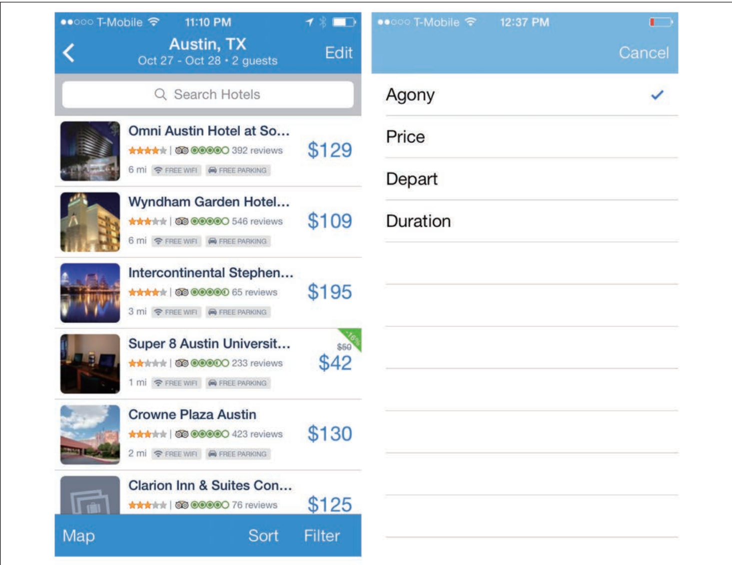
排序表单模式可能需要用户做出更多操作——打开表单，选择一个选项，而且可能还要点击“完成”“应用”或是“返回”按钮。

图 4-43
iOS 版 HomeAway




Hipmunk 在安卓平台上使用了排序浮层模式，但在 iOS 平台上却选择了排序表单模式。一个排序选项在被点击后立即执行，无需点击“应用”或“完成”按钮。

图 4-44
iOS 版 Hipmunk



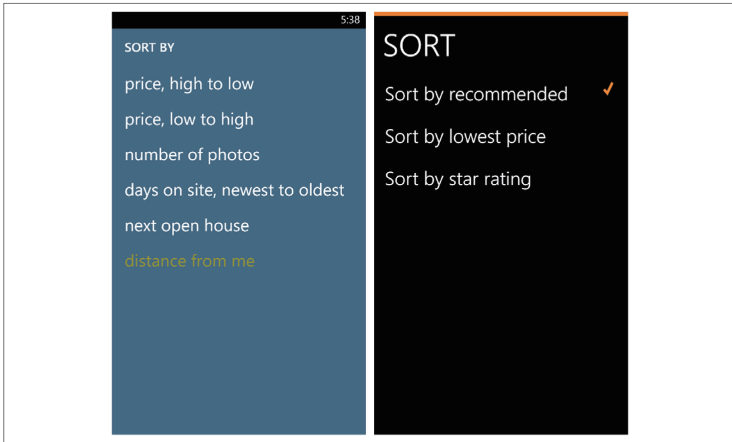



图 4-45
Windows Phone 版 Realtor.com 和 Kayak

遗憾的是，在 Windows Phone 平台的应用中，排序表单模式似乎占据了主导地位，而 Newegg 则采用了排序浮层模式，成为一个颇受欢迎的例外。Windows Phone 应用的另一种选择是利用透视控件，如下面的 Travelocity 示例所示。

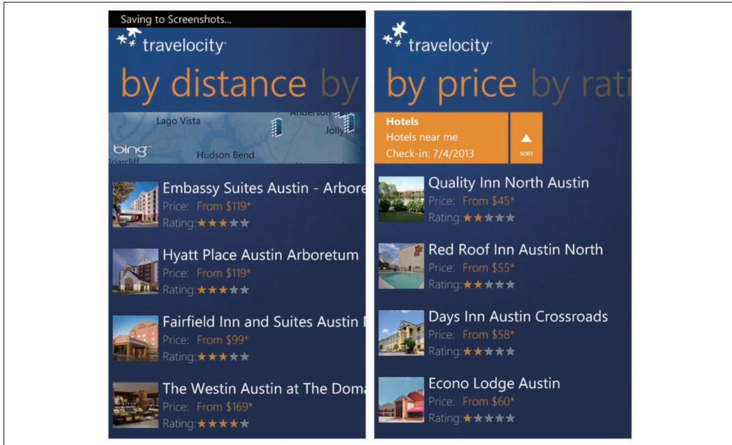



图 4-46
Windows Phone 版 Travelocity：用透视控件作为排序选项

一些应用将排序和筛选模式集成在一个表单中。这样做有一定的道理，因为两者都是用于进一步精细化搜索。但在 iOS 版 Priceline 和 Yelp 应用中，那样做会导致页面太长且过于拥挤，变成反模式案例（参见第 11 章）。Windows Phone 版的 Yelp 和 iOS 版 Foursquare 的单页设计就干净得多。

［小贴士］

在采用排序表单设计前，先考虑更高效的单页排序或排序浮层模式。如果要将排序模式和筛选模式集成到一页表单里，要尽可能保持页面简短、整齐。

图 4-47

iOS 版 Priceline 和 Yelp

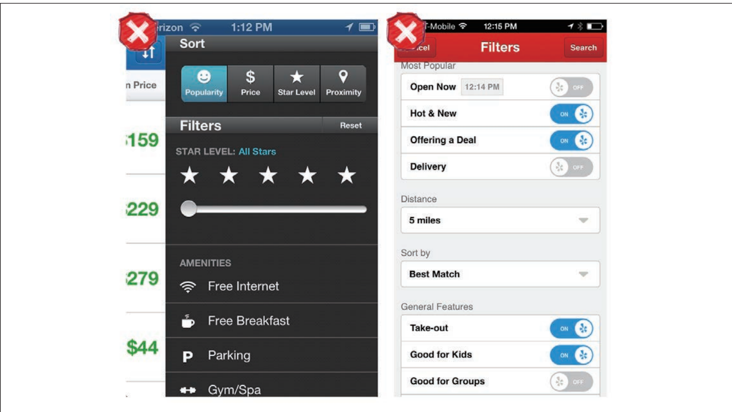
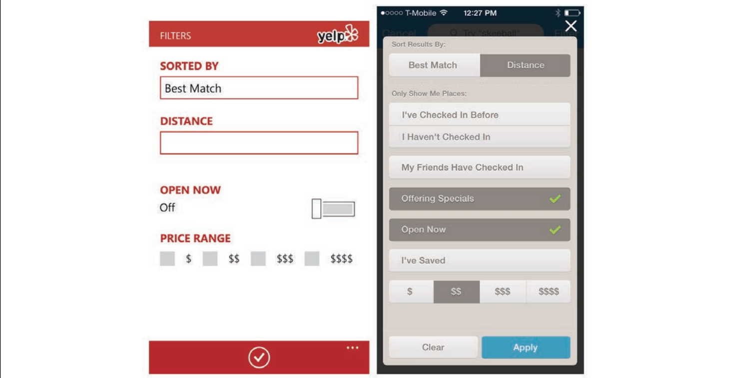



图 4-48

iOS 版 Foursquare 和 Windows Phone 版 Yelp




# 筛选模式

大型数据集可能因太难处理而变得几乎没用。为了让数据有意义，用户经常需要对结果进行筛选或细化。筛选器允许用户选择条件来缩减数据集，只留下对于他们来说最可用的、最相关的搜索结果。常见的筛选模式包括：

- 单页筛选
- 筛选浮层
- 筛选表单
- 筛选抽屉
- 基于手势的筛选

关于筛选技术的更多内容，请参看前面的搜索模式：限定范围的搜索。

## 单页筛选

与单页排序一样，单页筛选显示在搜索结果页。点击后，筛选立即执行。

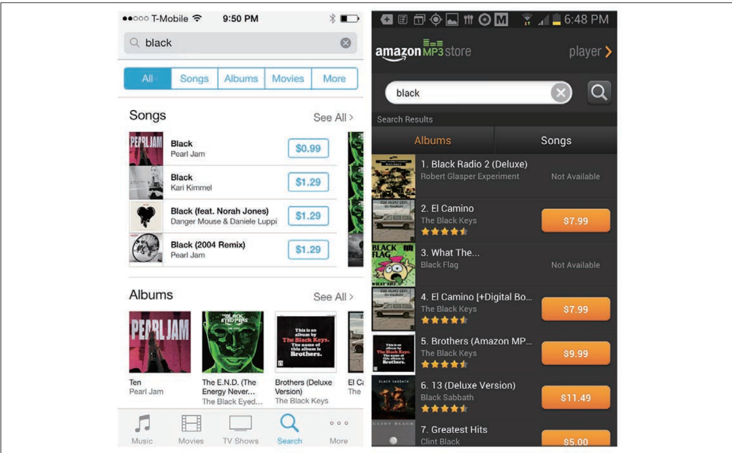



图 4-49
iOS 平台上的 iTunes 和安卓版 Amazon Cloud Player：位于顶端的单页筛选

安卓版谷歌应用在屏幕底端显示三种最常用的单页筛选。点击“More”，显示其他的筛选选项。当用户滚动屏幕浏览结果时，筛选隐藏起来，以便显示更多的搜索结果。

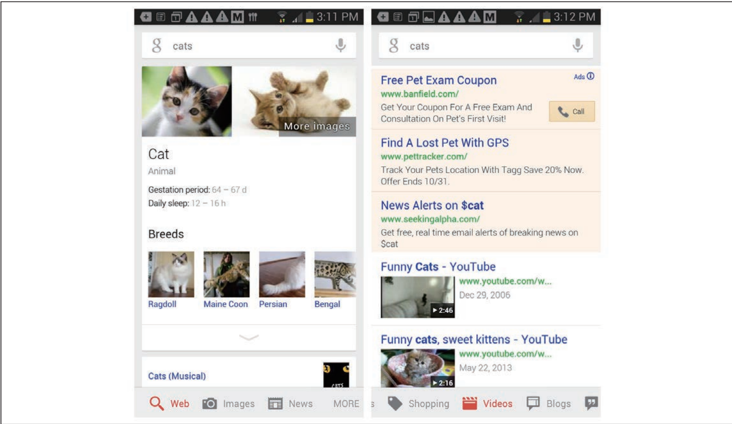



图 4-50
安卓版谷歌：置于底端的单页筛选

安卓版的 Groupon 和 Play Store 都使用了一行横向滚动筛选栏，分别让用户能够快速找到某一类型的优惠券和应用。安卓平台把这种控件称为滚动栏，而 Windows 平台则称之为透视项目标题（Pivot Header）。

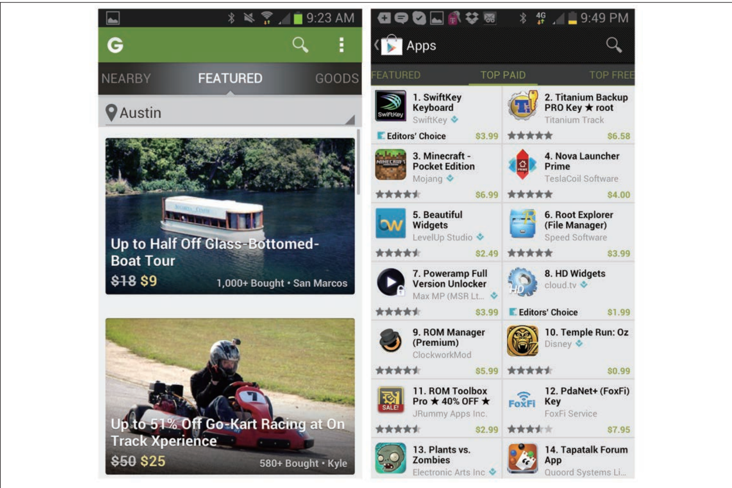



图 4-51
安卓版 Groupon 和 Play Store：用滚动栏实现筛选

SXSW GO 应用有两行筛选，用来选择日期、会议类型以及其他高级筛选选项。你可以看到高级筛选在何时处于开启状态，因为当筛选从关闭状态切换到开启状态时，颜色和文字都会改变。

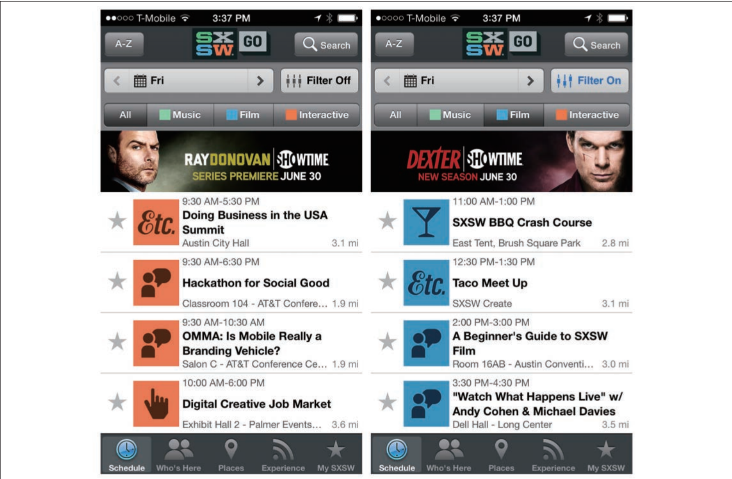



图 4-52
iOS 版 SXSW GO：双层筛选

在安卓应用中，下拉菜单控件可以很好地实现筛选列表。虽然不能一眼看出全部筛选选项，但它可以清晰地显示出当前被选中的选项。

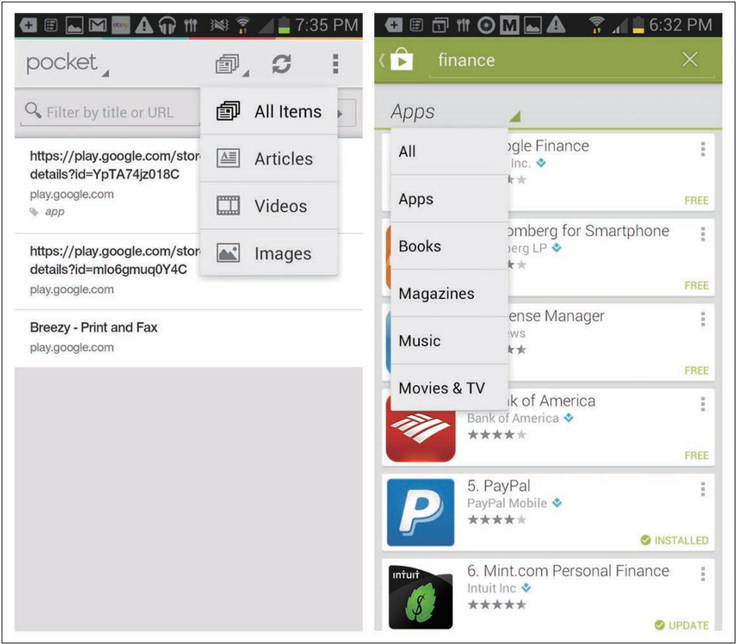



图 4-53
安卓版 Pocket 和 Play Store：下拉菜单控件筛选

［小贴士］

筛选选项的文字应该清晰易懂。可能的话，提供筛选器是否被应用或是否开启的视觉线索，特别是在筛选控件有时会隐藏起来的情况下。

## 筛选浮层

筛选浮层模式是除单页筛选外的另一种可行的方案。这种模式不会占用宝贵的屏幕空间，而且如果一个更长、更明确的标签效果更好的话，你也不必非用短标签不可。用户可以点击“筛选”（Filter）或“设定条件”（Refine）标签开启浮层，也可以使用能够清晰表达可以应用筛选条件的图标。

图 4-54

安卓版 Airbnb：筛选浮层

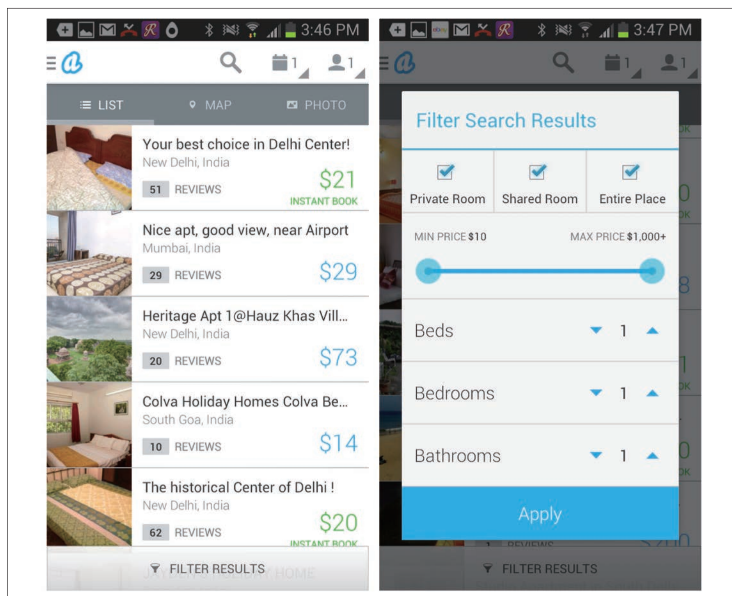
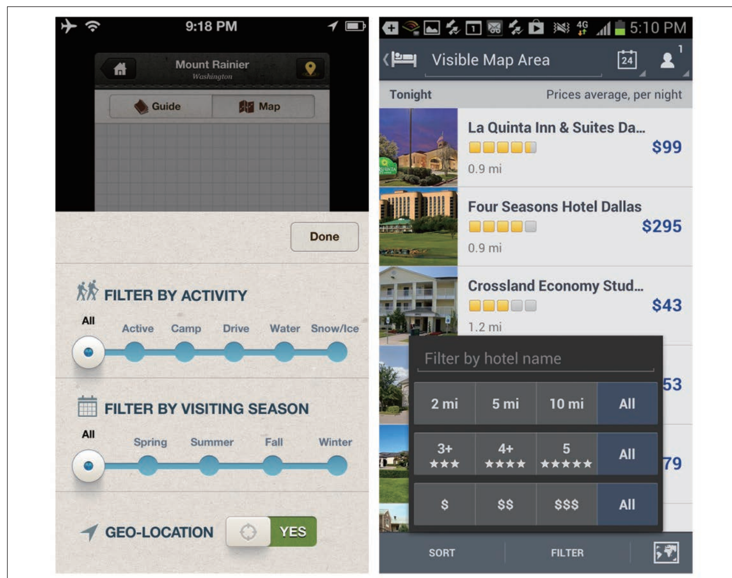



图 4-55

iOS 版 National Parks 和安卓版 Expedia：筛选浮层




［小贴士］

筛选浮层既能节省屏幕空间，又能包括比单页筛选更多的高级筛选选项。但筛选浮层需要消耗用户更多的时间和精力，所以在选择这种模式前要权衡利弊。

Ness 在标题栏上用图标显示可供选择的三种筛选条件，每一项都有各自不同的浮层。筛选选项显示在分段控件的下方。

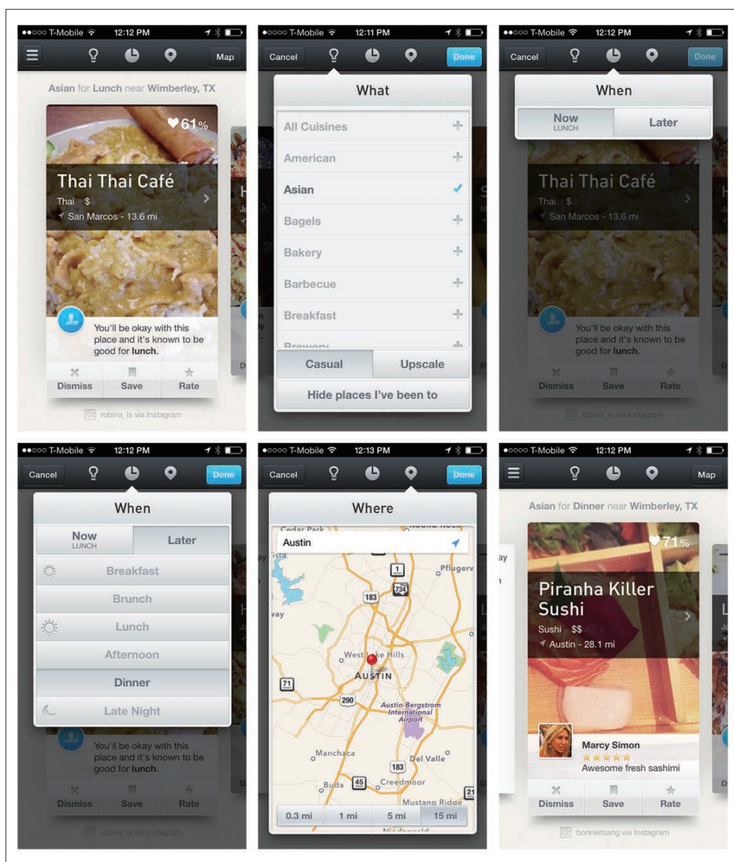



图 4-56
iOS 版 Ness：每个筛选选项都有筛选浮层

## 筛选表单

筛选表单模式也需要用户做出更多操作——打开表单，选择一个或多个选项，然后点击“应用”按钮。筛选表单很适合筛选选项很多的情况，它可以将这些选项分组显示。

筛选表单不可避免地要暂时隐藏被操作的搜索结果。注意 iOS 版 Zillow 是怎样在用户做出选择时动态地更新搜索结果数的，这样的设计可以帮助用户判断何时可以对搜索结果进行操作。

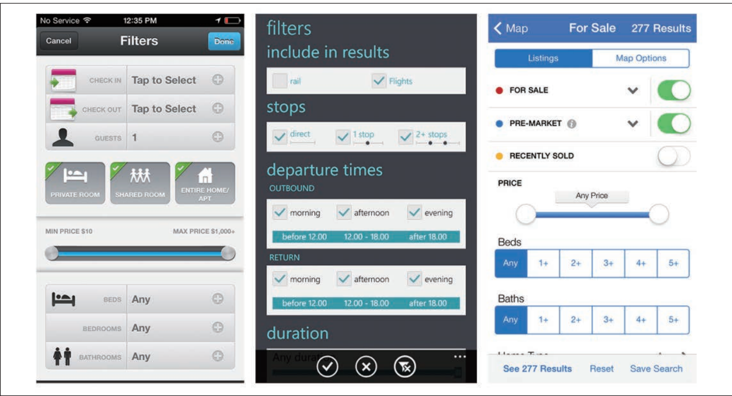



图 4-57

iOS 版 Airbnb、Windows Phone 版 Skyscanner、iOS 7 版 Zillow：筛选表单

筛选表单也适用于嵌套式或层级式的选择项，如 iOS 7 版 Walmart 应用所示。

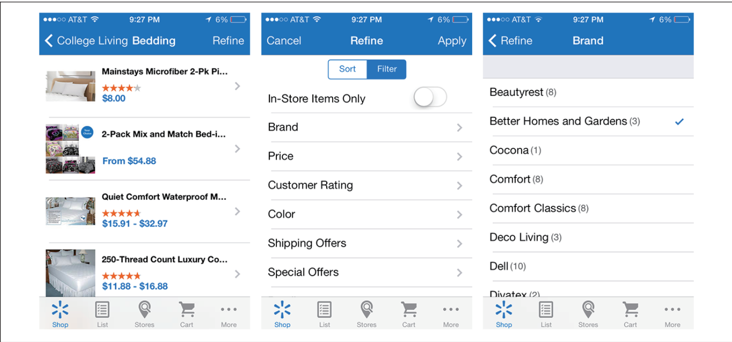



图 4-58

iOS 7 版 Walmart：嵌套式 / 层级式的筛选表单

［小贴士］

同搜索表单一样，应当尽可能简化筛选表单，遵守系统表单设计规范。不要设计新型的、自以为巧妙的筛选控件，这样可以避免设计成反模式（见第 11 章）。

Windows Phone 版 Allrecipes 提供了与搜索相关的筛选表单，类似于本章前面介绍过的限定范围的搜索模式 。

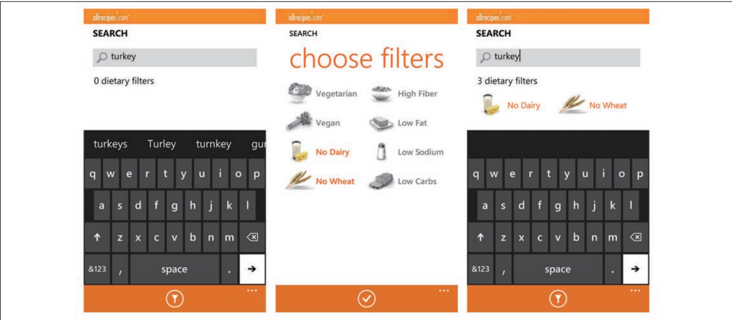



图 4-59

Windows Phone 版 Allrecipes：与搜索相关的筛选表单

## 筛选抽屉

筛选抽屉在某些方面与筛选浮层和筛选表单非常相似。像筛选浮层和筛选表单一样，筛选抽屉也需要额外点击“应用”或“完成”按钮以获取筛选结果。它们之间的主要区别是，筛选抽屉是从当前页面侧边滑出的，这样允许用户在筛选的过程中停留在筛选结果页上。理想情况下，筛选抽屉会把搜索结果显露出来一点。

在 iOS 版 Target 应用中，筛选抽屉向下滑出，完全盖住了搜索结果。当抽屉关闭时，没有指示表明被执行的筛选项。筛选处于活动状态的唯一反馈是，“filter”标签显示为蓝色。

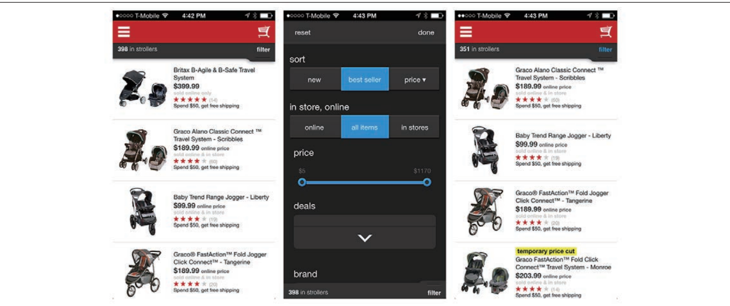



图 4-60
iOS 版 Target：筛选抽屉

以下是一些遵循筛选抽屉最佳实践的改版示例：

- 打开和关闭抽屉是对应的操作
- 动态更新搜索结果的个数
- 搜索结果在背景层中始终可见
- 在筛选器被执行后给予清楚的反馈

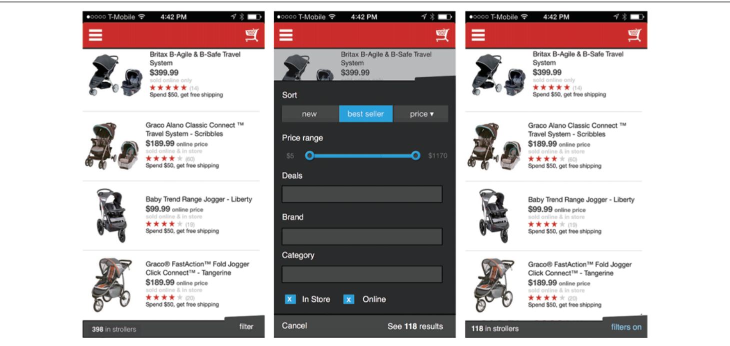



图 4-61
iOS 版 Target：改版后的筛选抽屉

iOS 版 ToDo 在导航抽屉中提供了筛选。这种设计只有在筛选很简单的情况下才适用，而且应该经过用户测试。如果用户更想打开一个列表，直接进行筛选，那么这种设计就不太有效了。

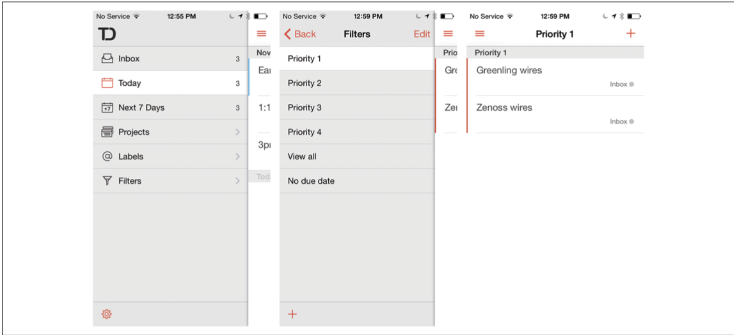



图 4-62
iOS 版 ToDo：导航抽屉中的筛选器

［小贴士］

筛选抽屉能让用户在筛选过程中停留在内容页上。在选项被应用时显示更新后的搜索结果个数，在抽屉处于关闭状态时提供筛选是否活动的视觉指示。

iOS 版 eBay 多层级的筛选抽屉是个很好的设计。在搜索结果页点击“Refine”，会从右侧滑出筛选抽屉，显示出所有可能的优化选项。

你也可以直接在搜索结果页上筛选特定的选项，比如类别、平台和购买版本，点击那些标签，筛选抽屉就会打开对应的层级。

搜索结果的个数始终显示在抽屉顶端，当抽屉关闭时，控制栏指示出当前应用的筛选项。可在视频中观看这种交互设计的效果。

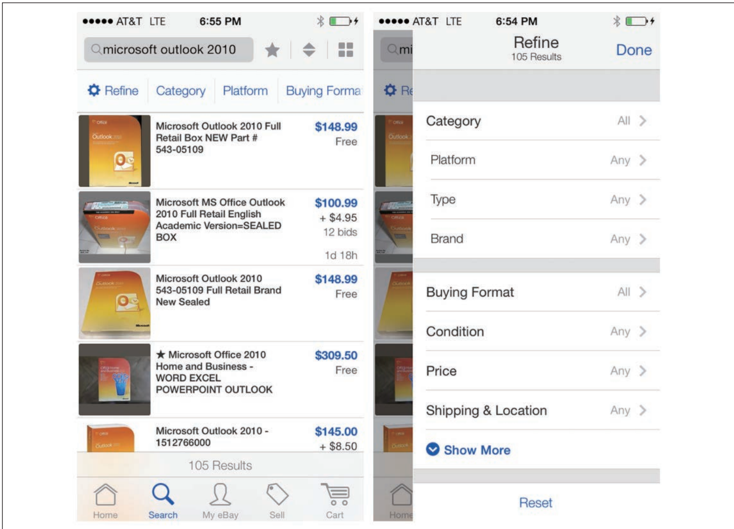



图 4-63
iOS 版 eBay：多层级筛选抽屉（http://www.youtube.com/watch?v=M_R3eSFcML0）

## 基于手势的筛选

当搜索结果的相关性取决于地理位置，比如在地图上、图表或图形中显示结果时，缩放、放大和拖动手势可以用于筛选结果。iOS 版 HomeAway 显示了特定半径区域的默认视图。双指张开放大地图以减小区域半径，然后显示结果；双指合并缩小地图以扩大半径，显示更多结果。Windows Phone 版的 Kayak 也提供了同样的功能。

> ［小贴士］
>
> 当结果的相关性取决于用户所处的地理位置时，基于手势的筛选就为用户提供了一种直观的方式来设定搜索结果的范围。可以把这种模式和其他筛选模式结合起来，因为地理位置可能不是用户唯一想要的筛选条件。

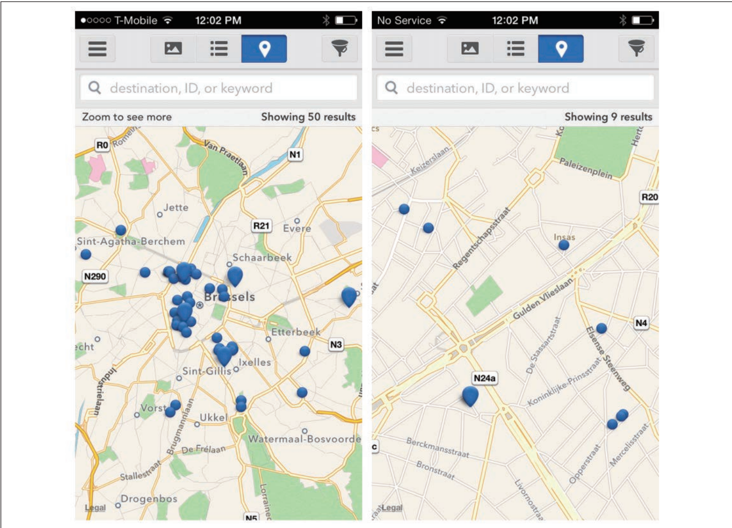



图 4-64

iOS 版 HomeAway：基于手势的筛选

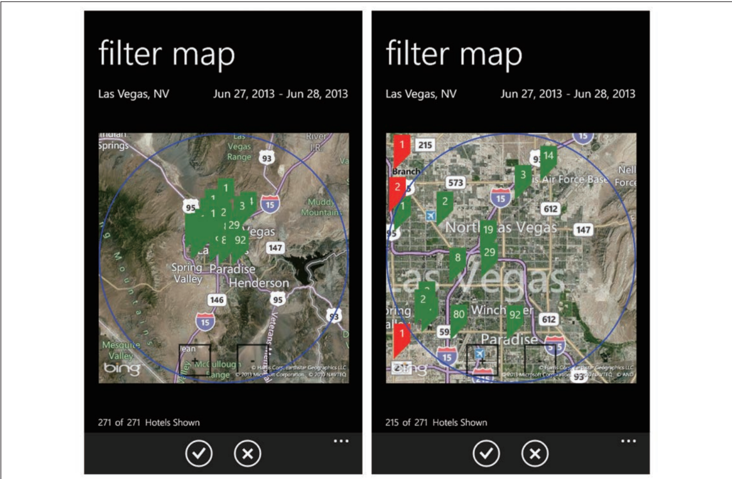
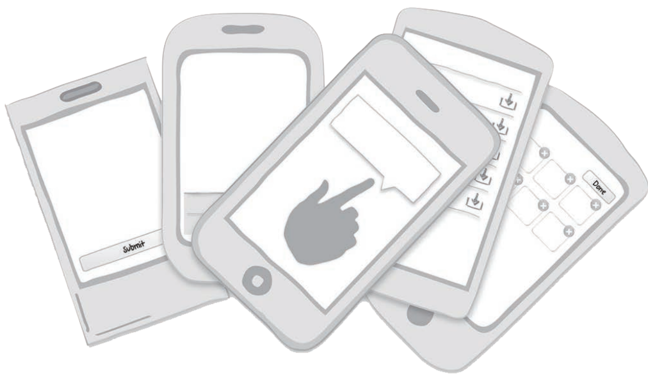



图 4-65

Windows Phone 版 Kayak：基于手势的筛选

# 第 5 章

# 工具




- 工具模式

  工具栏（Toolbar）、工具箱（Toolbox）、行为召唤按钮（Call to Action Button）、行内操作（Inline Actions）、多状态按钮（Multi-State Buttons）、上下文工具（Contextual Tools）、批量操作（Bulk Actions）、锁屏控件（Lock Screen Controls）

在第 1 章中，我们讨论过，由于屏幕尺寸有限，设计师们不得不思考“突破画布”，探索新的解决方案。在本章正式开始前，我鼓励你接受另一项挑战——摆脱三十年来抽象的桌面端隐喻，开始真正地面向触屏界面设计。为了启发灵感，我们先从 Josh Clark 的“按钮只是一种变通手段”（Buttons Are a Hack）这一言论开始（http://globalmoxie.com/blog/buttons-are-a-hack.shtml）：

> **按钮只是一种变通手段。**在现实世界中，按钮往往必不可少，但它们和操控对象有一定的距离，它们是操控主体的辅助工具。比如在此处按下开关打开的是别处的灯。人们必须通过学习才能进行这些间接的交互，且它们之间的关联不是很明显。触屏设备引发了一场变革，我们能够越来越多地将主体内容作

为控制对象，从而制造出直接交互的幻觉。

我并不是建议将我们熟悉的按钮统统抛弃。毕竟，电灯的开关应该保留，按钮也是如此，尤其是在有必要触发抽象操作的情况下（比如“分享到 Twitter”）。但是知道这些设备的用途非常重要：在不可能进行直接交互时才需要使用。触屏界面允许用户在更多的情景中进行直接交互。随着新型解决方案的出现，我们应该把久经考验的变通方法放在一边。在设计触屏交互时，应当不断问自己：我真的需要在这里加上一个按钮或控件吗？

让我们看看 TweetCaster 和 Kayak 的设计示例。在 TweetCaster 中，主界面是一个推文列表。点击一个推文，就会在新页面将其打开。要想切换至上一条或下一条推文，你需要使用导航栏中的两个按钮（向上和向下的箭头）。但我们真的需要按钮来切换至下一项或前一项吗？还是说，这样的设计只是桌面设计遗留下来的产物？在触屏上，滑动翻页是一种更有效、更自然的切换内容的方式，关于这一点我会在本章后面的部分加以说明。

那么 Kayak 地图上的“加 / 减号”放大镜又怎么样呢？我记得从 20 世纪 90 年代我拥有第一台电脑起，这个东西就存在了。丢掉这个放大缩小按钮吧，你完全可以用双指缩放手势实现这一功能。




图 5-1

iOS 版 TweetCaster 和安卓版 Kayak：过时的桌面端隐喻设计

我相当确定，到本书下一版问世时，富有创意的新触摸式工具模式将出现。现在，我们只讨论当前最普遍的设计模式。诚然，这些模式中有一些是传统桌面端隐喻的延伸——只是它们的触发方式不是鼠标点击，而是触摸。但另外一些模式则融合了触摸反馈，这些方式是桌面端界面所做不到的。

# 工具栏

工具栏是提供页面特定操作的最常见的方式。对于工具栏，每个操作系统都有自己的术语和设计指南。

## iOS

iOS 人机界面指南（https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/Bars.html#//apple_ref/doc/uid/TP40006556-CH12-SW4）中写道：

> 尽管工具栏看上去和导航栏或标签栏很像，但它并不能用来导航。相反，工具栏使用户可以在当前页面操作内容，它始终显示在页面或 iPhone 屏幕的底端。

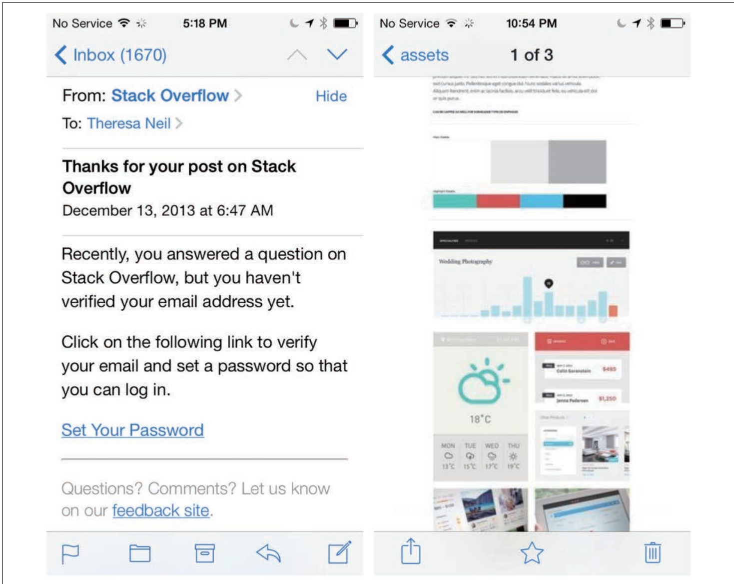



图 5-2
iOS 版 Mail 和 Dropbox：不要混淆工具栏和导航栏

Expedia 在工具栏的基础上设计了级联菜单，而 Wunderlist 为节省屏幕空间设计了一个可收起的自定义工具栏[1]。

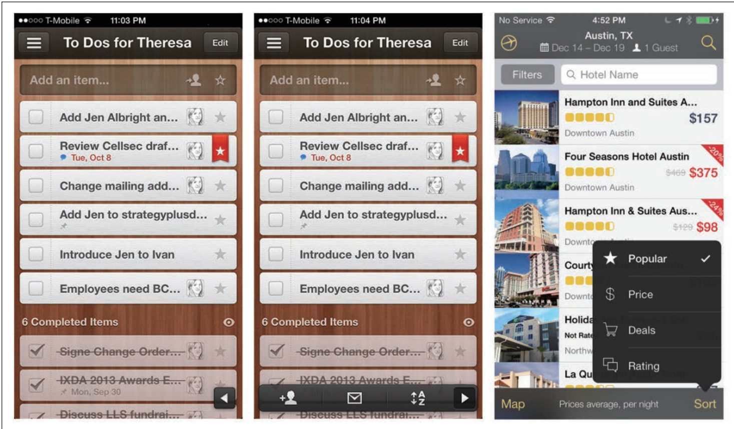



图 5-3
Expedia 和 Wunderlist 的自定义工具栏

另一种设计方案是，一个操作按钮可以触发一页操作菜单（Action Sheet），上面显示了选中按钮的其他可选项。比如在 Narrato 中，点击照片按钮会打开一个标准的操作菜单，用户可以选择照片的来源，而点击笑脸按钮会弹出自定义操作菜单，用户可以选择心情。这一巧妙的设计很好地体现了 iOS 设计指南的精神，而不是照搬其条文。

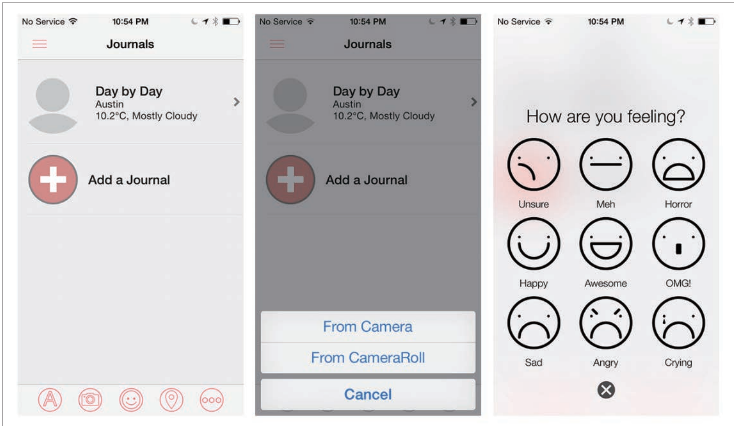



图 5-4
iOS 版 Narrato：两种不同的操作菜单设计

注 1：最新版 Wunderlist 已经改成固定工具栏。（译者注）

# 安卓

Jelly Bean 的发布为安卓界面带来了一些新的变化（http://developer.android.com/design/patterns/actionbar.html）。安卓系统不像 iOS 系统那样有一个专门的工具栏，它的操作按钮放置在操作栏的右上角，而操作栏位于页面顶端，通常固定在应用的每个页面上。

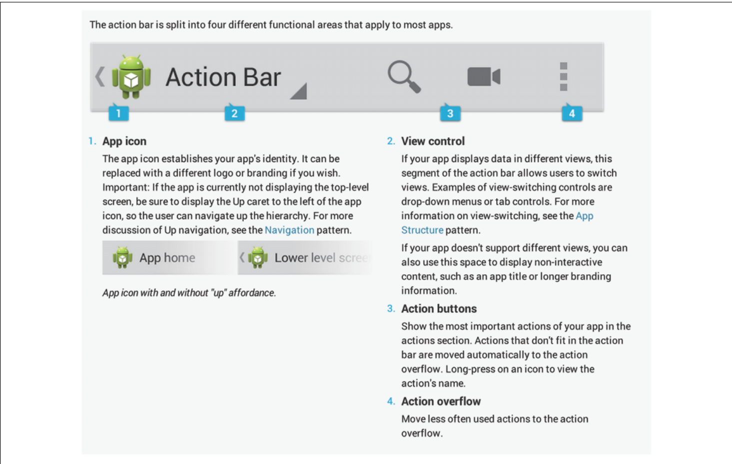



图 5-5

将操作栏分隔成四个功能区域，适用于绝大多数应用

操作按钮应该针对当前页面的情境发生变化，如 Gmail 应用所示。主要操作按钮应该显示在顶部操作栏上，次要操作应该放进溢出菜单（Overflow Menu）里，用户可以通过点击垂直的省略号图标进行这些操作。

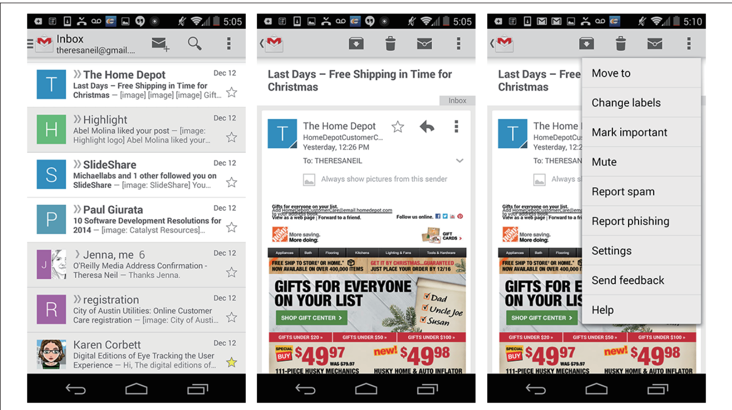



图 5-6

安卓版 Gmail：操作栏按钮随页面情景发生变化

安卓系统的一个很好的特点是，长按一个操作按钮会出现工具提示，说明这个按钮的用途。一个不太好的特点是，操作栏里不仅有操作按钮，还不得不放进所有必要的导航控件（标题名称、抽屉图标、向上返回按钮、下拉菜单等），因此操作栏很快就会变得拥挤不堪。

## Windows Phone

Windows Phone 应用中的工具栏被称为应用栏（Application Bar，http://msdn.microsoft.com/library/windows/apps/ff431813(v=vs.105).aspx）。主要操作以带有文字标签的图标显示出来，次要操作（称为菜单项）可以折叠进省略号图标。

当看到亚马逊应用试图把应用栏作为主导航控件（看看每页底端的“首页”“搜索”和“购物车”图标吧）时，我非常惊讶。这让我不禁认为其产品设计是由 iOS 狂热爱好者所驱动，而不是由一个熟悉 Windows Phone 设计规范的团队所打造。

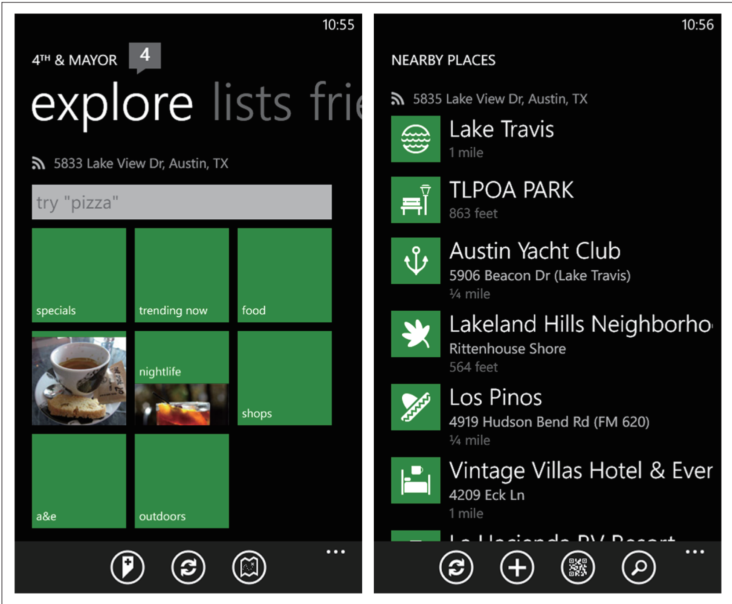
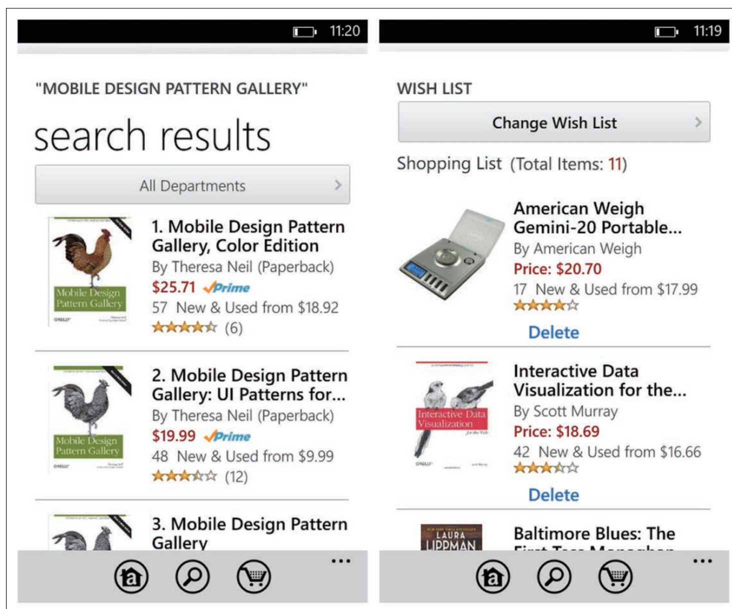



图 5-7

Windows Phone 版 4th & Mayor：放置主要操作的应用栏

图 5-8

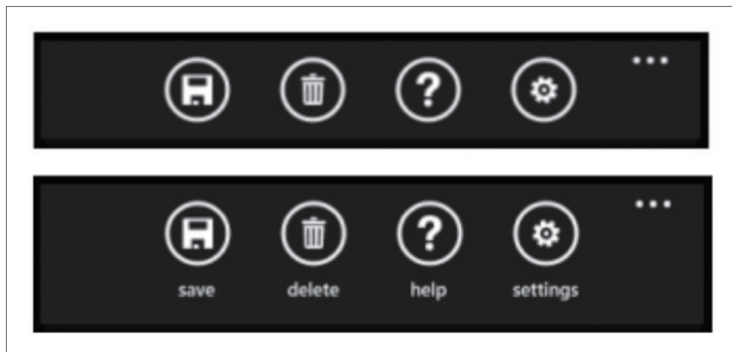
Windows Phone 版亚马逊：错误地将应用栏作为主导航




和安卓系统一样，Windows Phone 系统也提供显示图标标签的方式。点击省略号图标，应用栏的高度就会增加一点，以便在图标下方显示其文字标签。

图 5-9

Windows Phone 系统的应用栏：点击省略号显示标签




［小贴士］

在工具栏的放置和行为方面，要遵守针对特定系统的建议。选择用户熟悉、易识别的图标。

## 与操作系统无关的模式：上下文工具栏

上下文工具栏也适用于输入栏或输入框中使用的操作按钮，比如 Twitter、Evernote 和 OneNote 工具栏中的格式化和添加附件选项。

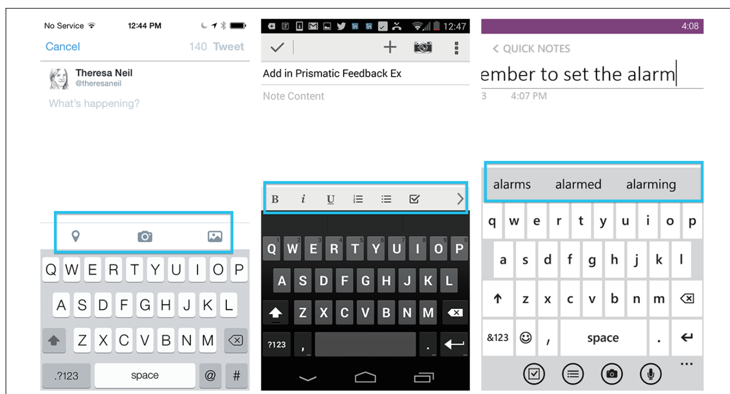



图 5-10
iOS 版 Twitter、安卓版 Evernote 和 Windows Phone 版 OneNote：用于文本输入的上下文工具栏

# 工具箱

工具箱类似于增强型的工具栏。这种模式在提供大量效果处理工具的摄影和设计类应用中最为常见。

通常情况下，主工具箱放置在屏幕底端。选项可能超出屏幕宽度，需要通过滑动选择选项。从主工具箱中选择一个工具，会显示该工具特定的一个新的选项集。比方说，在 Pixlr 中，选择“调整”（Adjustment）就会显示全部可选的调整选项，比如“亮白”（Whiten）、“修补”（Touch Up）、“焦点”（Focal Blur），等等。



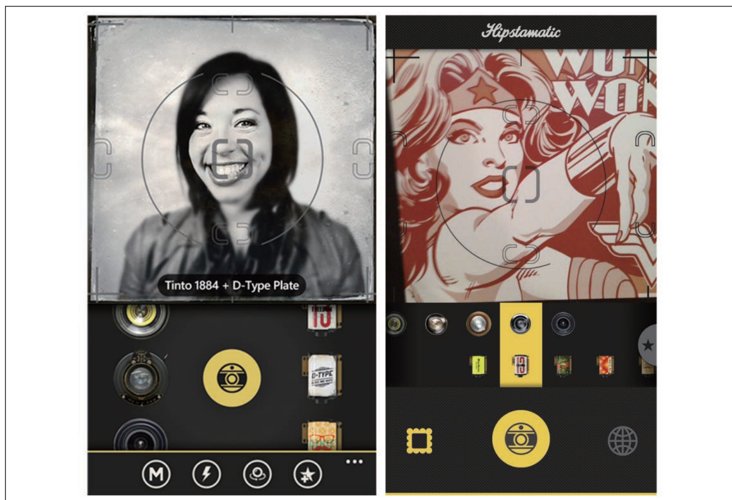

图 5-11
安卓版 Pixlr：在工具箱里选择一个工具后，会显示该工具的选项

Repix 和 Oggl 采用了相同的模式，但没有使用浮层选项菜单，而是在工具箱上方增加了一行，以显示所选中工具的各个选项。

图 5-12
iOS 版 Repix：工具选项作为第二行显示在工具箱上方




图 5-13
Windows Phone 和 iOS 版 Oggl：不同系统的工具箱设计有所区别




**[小贴士]**

标签加图标的形式可以增强工具栏和工具箱的可用性。如果工具选项把主工具箱隐藏起来，那么应当使用一些方法（比如页面标签）让用户知道他们所处的位置，而且要确保设计中有“取消”按钮。

还有一种实现方式是显示主要的工具集，当选择了一个工具时，其子选项才显示出来，隐藏原先的主选项。在 Aviary 示例中，首先选择“Orientation”，然后关于方向的各个选项才显示出来。使用这种模式时要记住两件事：因为主工具栏现在不可见，所以最好重新说明哪个菜单项正在生效；你应该为这种模式提供退出路径，比如“取消”按钮。

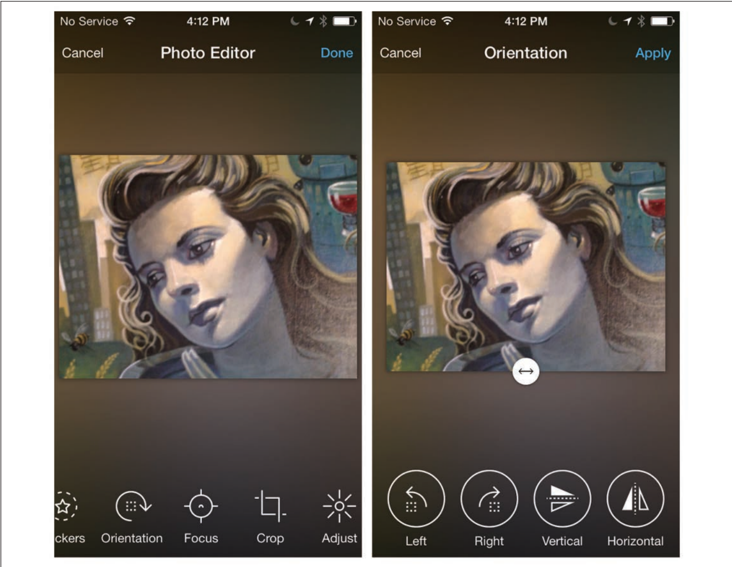



图 5-14
iOS 版 Aviary：右图为选定工具（Orientation）的选项

Virtual Makeover 使用了这种模式来展现多种不同的化妆工具。它的扇形工具箱出人意料地好用。

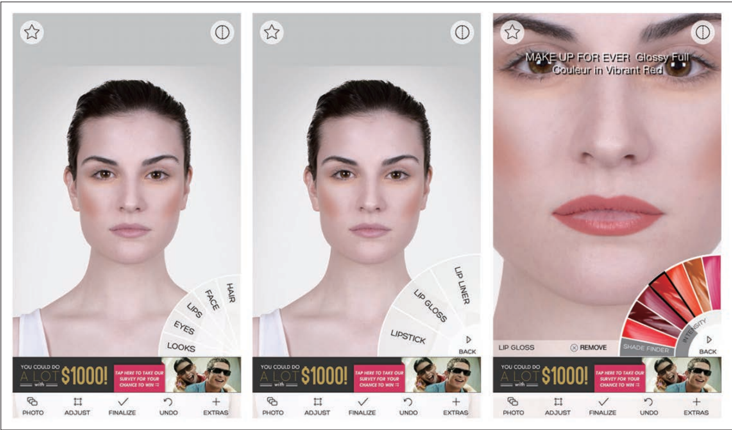



图 5-15
iOS 版 Virtual Makeover：扇形工具箱

## 行为召唤按钮

当你只在页面上显示单一操作的时候，“行为召唤按钮”可能比工具栏更好一些。Frank & Oak、Etsy 和 Hipmunk 就是这样的例子。

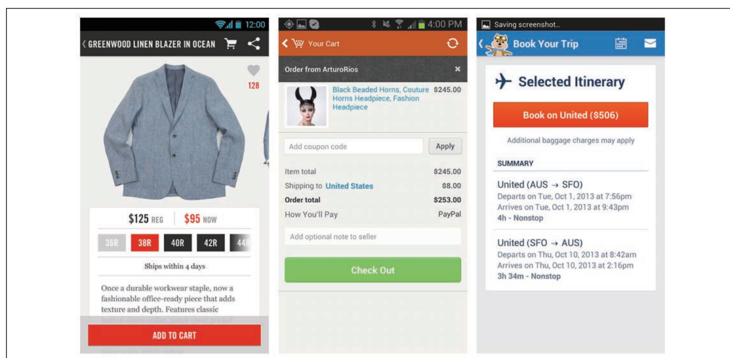



图 5-16
安卓版 Frank & Oak、Etsy 和 Hipmunk：突出的行为召唤按钮

这种模式也适用于有一个主要行为召唤按钮和一个次要行为召唤按钮的页面，比如 Foursquare 和 Fancy 的设计。要确保从视觉上能区分主要行为召唤按钮和其他按钮。

在这些例子中，行为召唤按钮是某个页面特定的工具，更重要的是，它是流程特有的工具。多年来设计师们都在纠结，应该在何处以及如何放置一个全局行为召唤按钮，也就是一个在应用的任何页面都可以点击进入的按钮。在第 1 章中，我们讨论了各类标准标签栏，其中包括标签栏中间位置放置单一操作按钮的设计模式，比如 Instagram 和 RunKeeperPro。

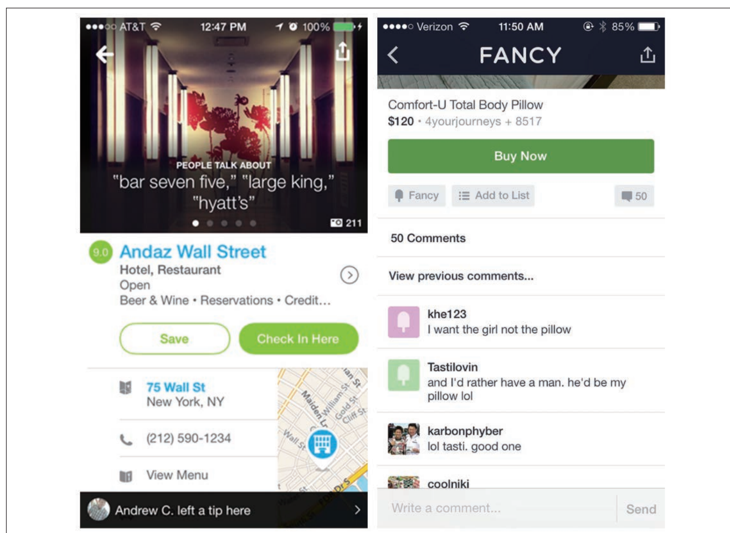



图 5-17
iOS 版 Foursquare 和 Fancy：主要行为召唤按钮与次要行为召唤按钮从视觉上区别开来

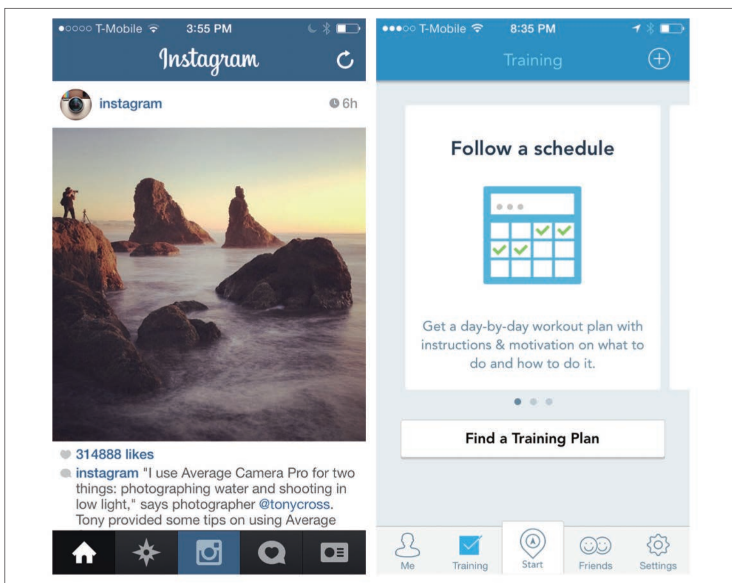



图 5-18
iOS 版 Instagram 和 RunKeeperPro：标签栏上的全局行为召唤按钮

iOS 6 版的 Tumblr 应用中，点击全局行为召唤按钮会弹出一组级联菜单选项。然而在 iOS 7 版的设计中，点击全局行为召唤按钮则打开全屏的选项。

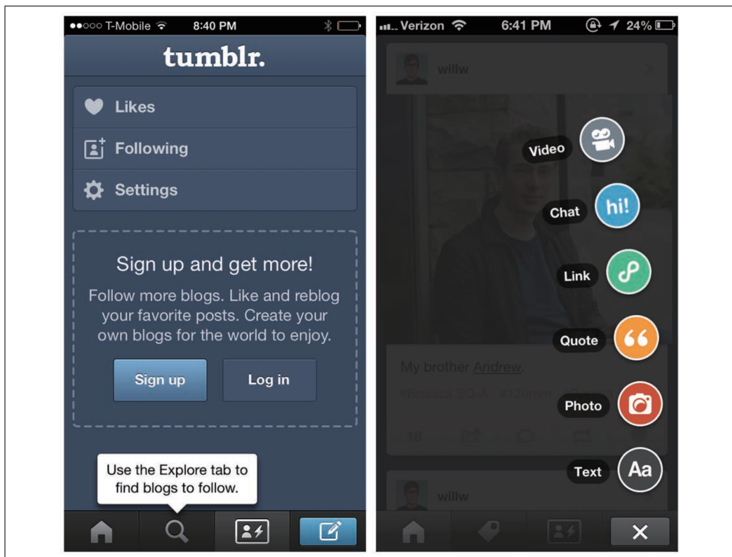
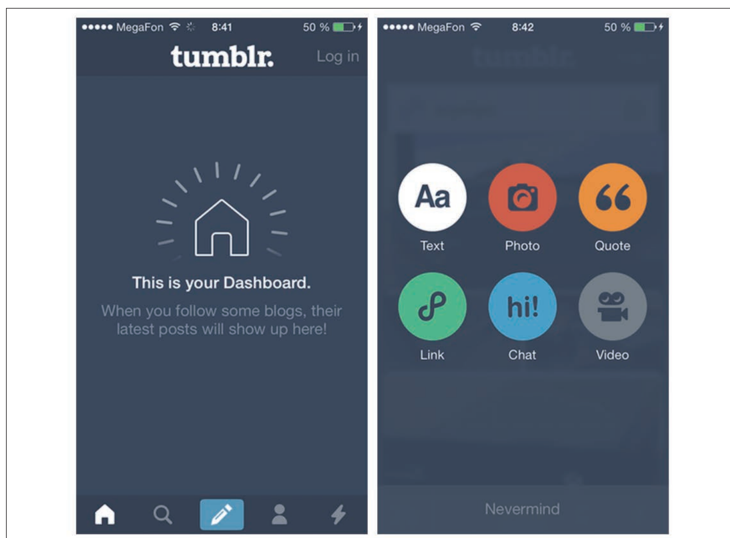



图 5-19
iOS 6 版 Tumblr：不平衡的右侧全局行为召唤按钮呈扇状弹出选项

图 5-20

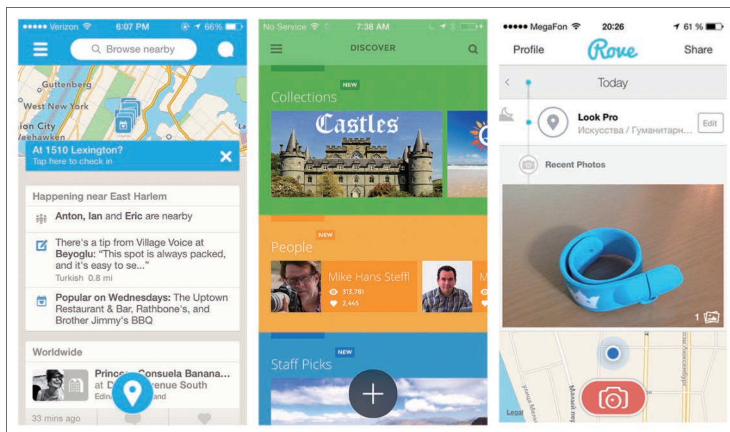
iOS 7 版的 Tumblr 把全局行为召唤按钮放在了中间位置，点击该按钮会弹出全屏操作选项




一个流行的趋势是，在屏幕底端放置一个悬浮的行为召唤按钮。Foursquare、Sphere 和 Rove 的悬浮行为召唤按钮就是这样的设计示例。

图 5-21

iOS 版 Foursquare、Sphere 和 Rove：悬浮行为召唤按钮




Path、Myspace 和 QuickBooks 中的悬浮式行为召唤按钮在点击后会显示更多选项。在这三个应用中，行为召唤按钮从符号“+”变成了“x”。顺便说一句，这种设计是对称交互的正确设计方式（比如“打开”和“关闭”按钮要放在同一位置）。

**［小贴士］**

在帮助用户成功地完成一个流程时，或让用户快捷使用有特定全局操作的应用时（比如拍摄照片、分享或录音录像入口），可以采用行为召唤按钮。

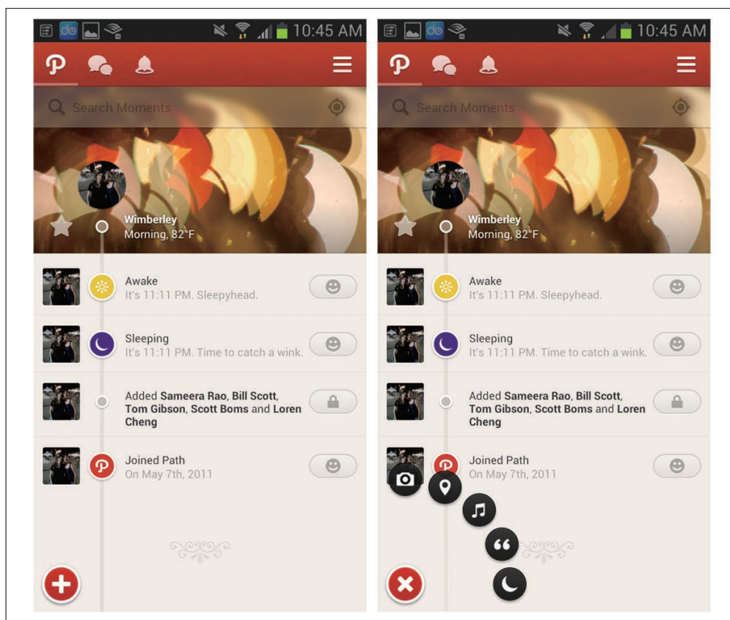



图 5-22
安卓版 Path：在点击悬浮的行为召唤按钮后，会显示浮动的选项

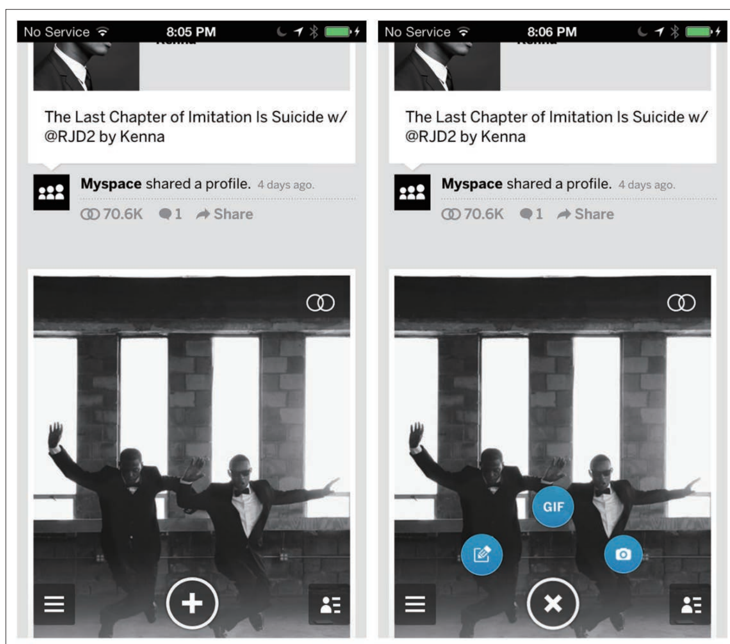
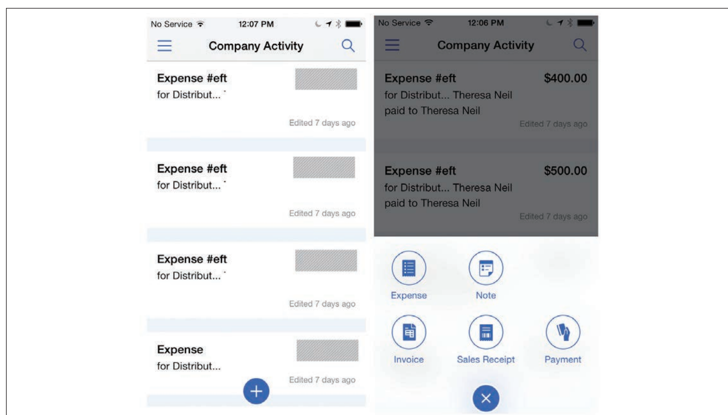



图 5-23
iOS 版 Myspace：注意，行为召唤按钮的图标在打开状态时从“+”号变为“x”号（即“关闭”）

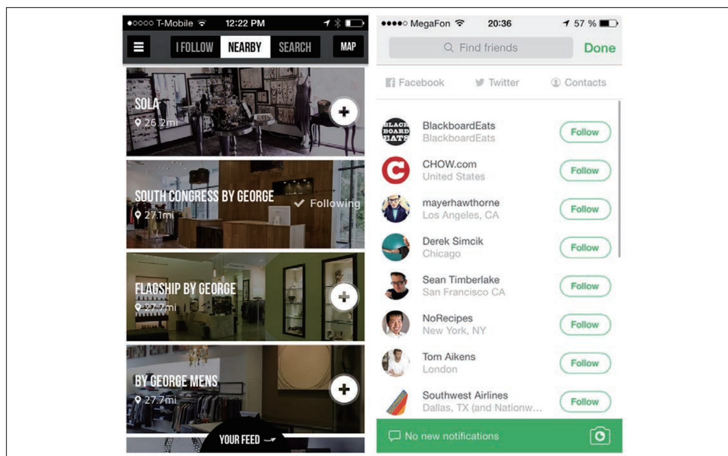
图 5-24
iOS 版 QuickBooks：打开 / 关闭浮动行为召唤按钮的另一种形式




## 行内操作

行内操作指的是操作按钮与其作用的对象在同一行，而不是在全局或整个屏幕层级上。比如，在 Nearby 和 Foodspotting 应用中，每一个列表项都有一个可显示两种状态（关注 / 已关注）的行内操作按钮。

图 5-25
iOS 版 Nearby 和 Foodspotting：可以显示两个状态的行内操作按钮




许多应用都依靠状态按钮来实现行内操作，比如 RetailMeNot 和 Gmail 中可以进行开关切换的星形按钮。参看下一个模式，“多状态按钮”，了解超过两个状态的实现。

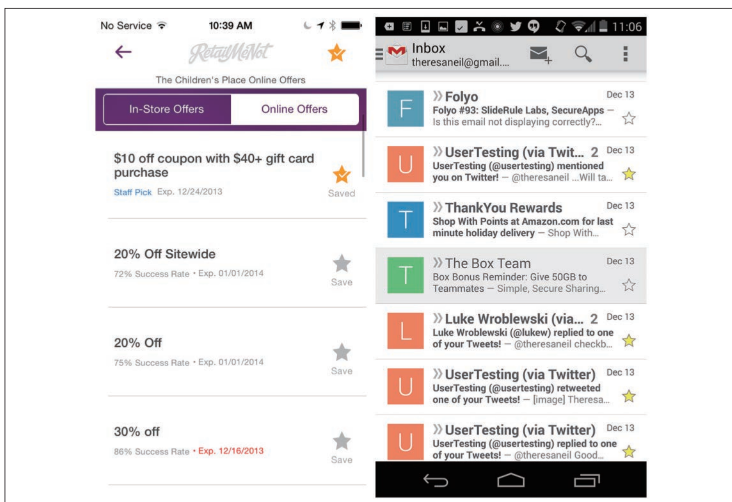



图 5-26

iOS 版 RetailMeNot 和安卓版 Gmail：不同状态的星形按钮

如 LinkedIn 和 Facebook 的截图所示，每一行中都有两个行内操作。如果你想在每一行中放置两个以上操作，可以考虑本章后面部分会讨论到的显示模式。

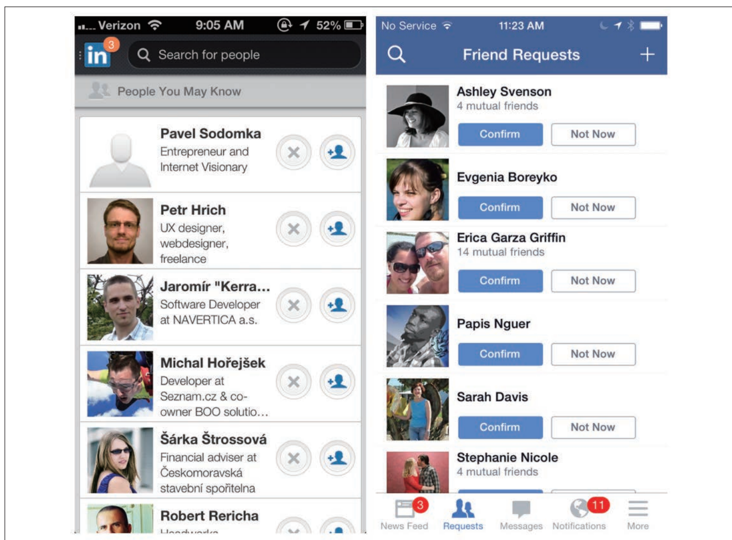



图 5-27

iOS 版 LinkedIn 和 Facebook：每一行显示两个行内操作

谷歌地图、Soundwave 和 Pulse 都在对象详情视图中提供了行内操作。

［小贴士］

行内操作应该靠近被操作对象。要选择用户熟悉的图标，并且可以考虑为图标加上文字标签。

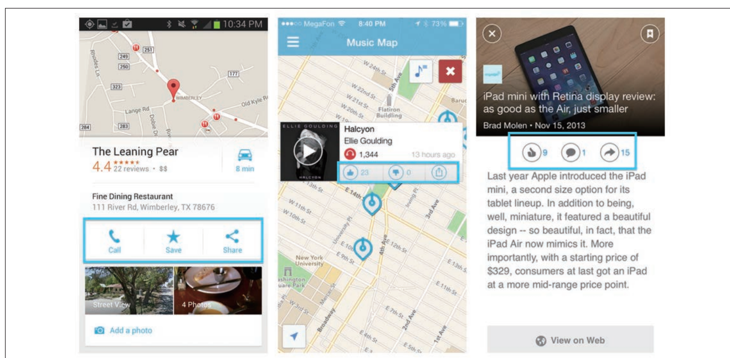



图 5-28

安卓版谷歌地图、iOS 版 Soundwave 和 Pulse：对象详情视图中的行内操作

## 多状态按钮

通常来说，你不应该在一个 UI 控件中设置多个操作，因为这会让页面过载，让用户感到有压力。多状态按钮是个例外。这种类型的按钮既用作操作触发器，同时也是一个反馈机制。

比如说，体验一下从 iOS 系统的应用商店中下载应用。首先，你会看到“购买”的行内操作。再次点击会显示“安装”。然后这个按钮变成了一个安卓进度指示器，最后，当应用下载完毕后，这个按钮显示为“打开”选项。更多例子请参看第 9 章。

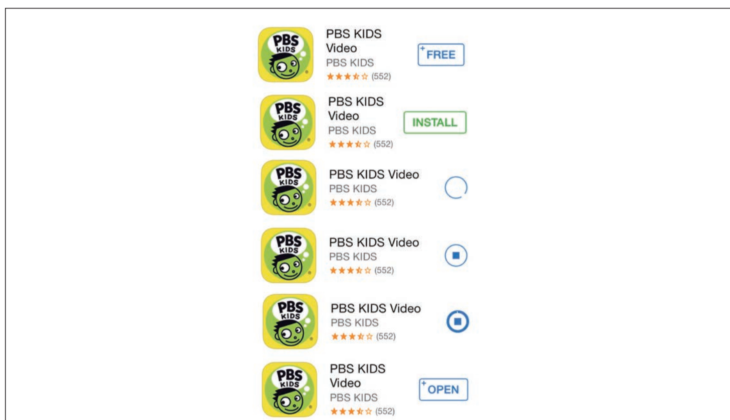



图 5-29

iOS 版应用商店：多状态按钮显示购买、安装、进度和打开选项

**［小贴士］**

在一系列紧密相关的过程被快速连续执行时，多状态按钮十分有效。在设计提示用户确认不可撤回的操作时，也可以考虑使用多状态按钮取代对话框。

多状态按钮的另一个常见用例是一种删除控件。在许多情况下，弹出删除确认对话框会打断用户的操作流程（参见第 11 章“愚蠢的

对话框”一节）。但在完成一项不能撤回的操作前进行提示确认是正确的做法，所以多状态按钮应运而生。（顺便提一下，归档电子邮件不是不可撤回的操作，但清空垃圾箱是不可撤回的操作。）

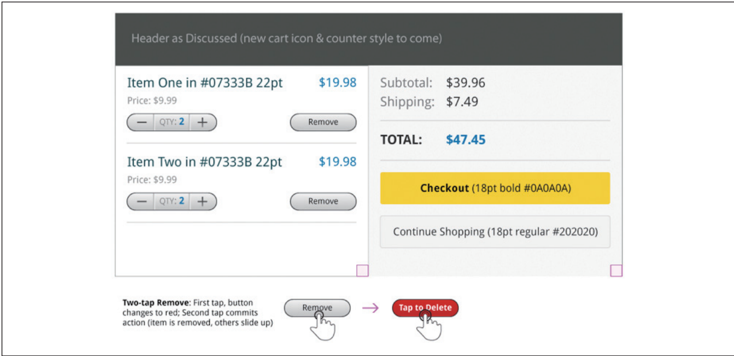



图 5-30
一个安卓应用的线框图和交互说明，使用多状态按钮实现“删除”操作

## 上下文工具

上下文工具专门针对一个对象或任务。只有在上下文建立起来之后才显示这些工具，这种设计很合乎情理（还能简化界面）。

比如，在 Luvocracy 应用中，点击任何一张图片中的省略号按钮，都会显示该图片的上下文工具。这种设计要比 Pinterest 的长按加滑动手势更容易被用户发现。当然，Pinterest 的按钮动画看上去很酷，但你可以观看视频，看看别人第一次试用它的情景。

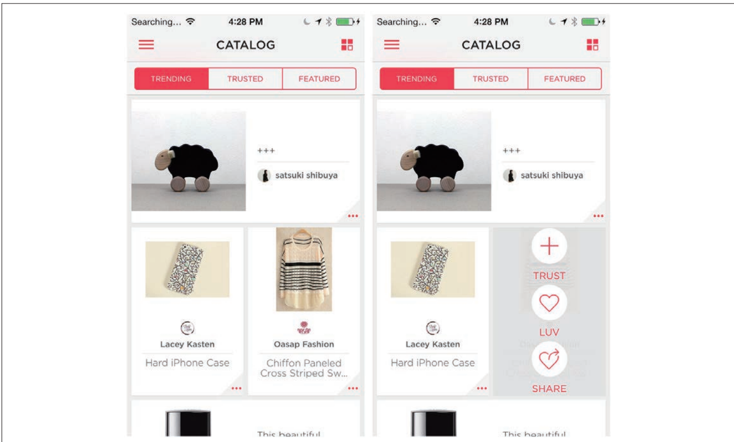
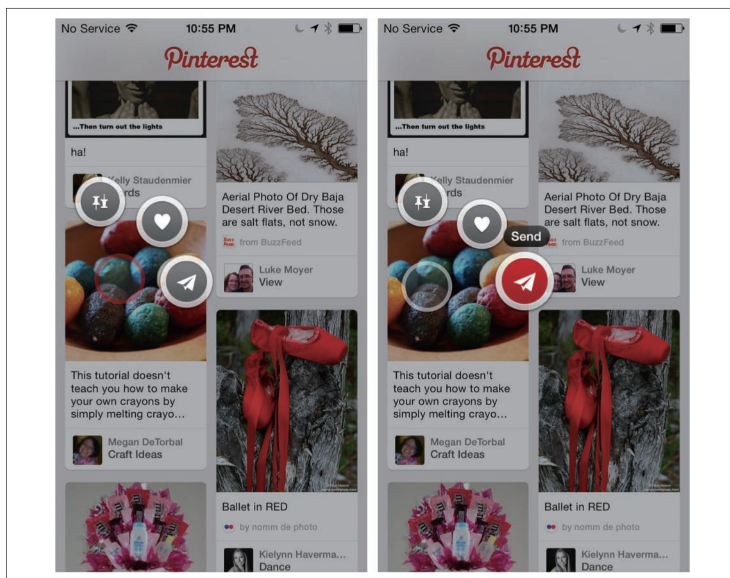


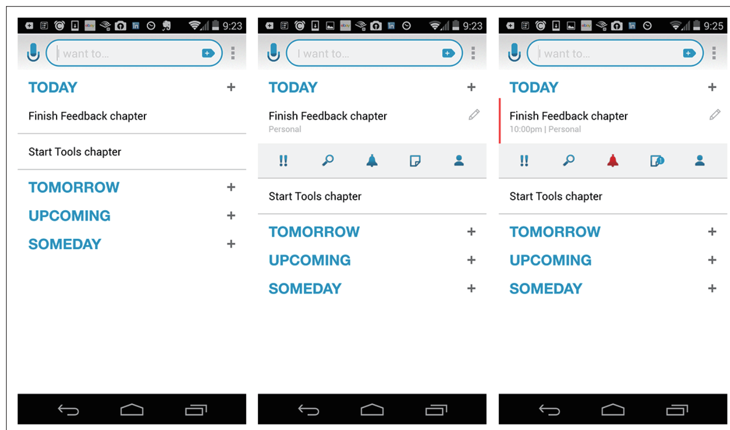

图 5-31
iOS 版 Luvocracy：点击省略号显示上下文工具

图 5-32
iOS 版 Pinterest：长按一个“收集”（Pin），然后按住不放并滑向一个按钮（http://www.youtube.com/watch?v=dvXYztMT-bI）




Any.do 也使用了通过简单点击来显示上下文工具的方式，它的工具包括设置优先级、添加提醒和分享。

图 5-33
安卓版 Any.do：点击显示上下文工具




要是列表项已经是触摸对象了，比如 Dropbox 的文件夹，怎么办呢？一个办法是增加一个“更多”图标，点击该图标，会显示上下文工具。Dropbox 使用了浮层来显示选项，但你也可以采用内嵌式把其他行推向下方。

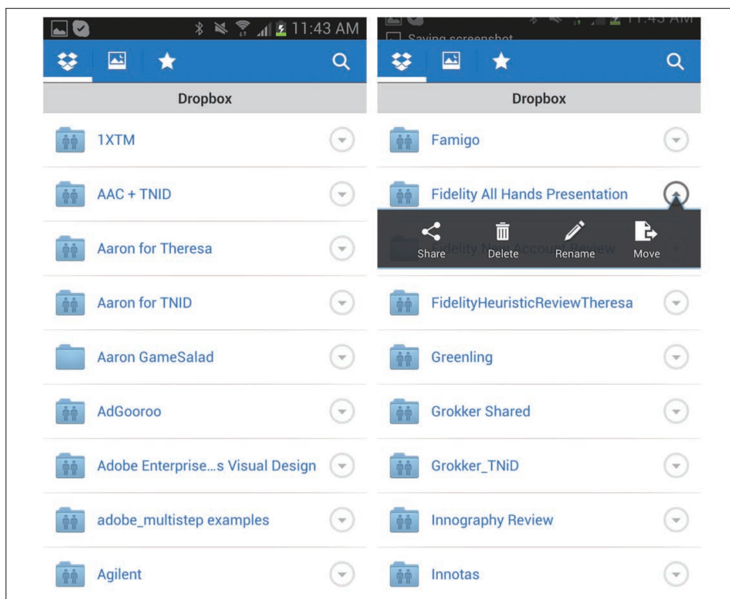



图 5-34
安卓版 Dropbox：点击“更多”图标显示上下文工具浮层

还有一种办法是用手势而不是点击来显示工具，比如滑动手势。Flixster 和 BaconReader 都通过水平滑动来显示上下文工具。

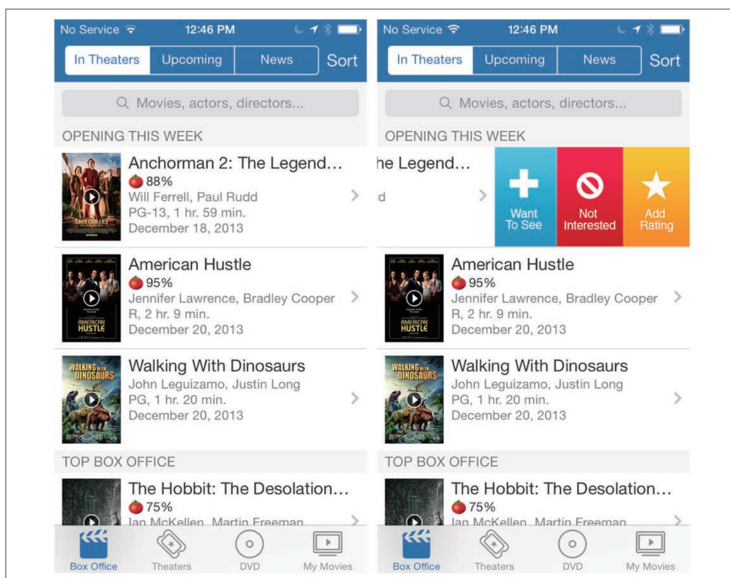
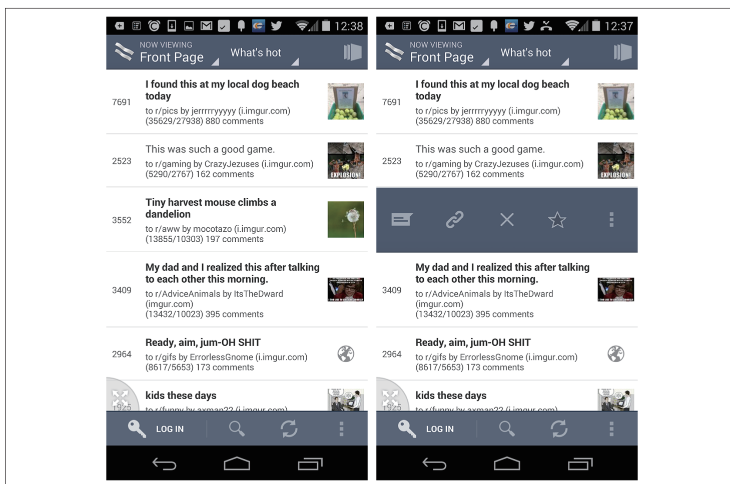



图 5-35
iOS 版 Flixster：左滑以显示上下文工具

图 5-36

安卓版 BaconReader：左滑或右滑以显示上下文工具




这种方式在效率类工具中非常流行，比如电子邮件和任务应用。但要注意，要设计一种直观的、有效的基于手势的界面并没有想象中那么容易。

［小贴士］

在选择一种手势来显示上下文工具时，应该先从最简单的解决方案着手，并对其可发现性和用户效率进行测试。

长按还可以用来选择一个对象，并呈现出其对应的上下文工具。在 Box 中，长按一个文件会触发该对象被选中的选择模式。可以选择更多对象，且上下文工具（下载、分享、复制、删除等）取代了操作按钮。要了解选择多个对象的其他方式，请参见下一模式——批量操作。

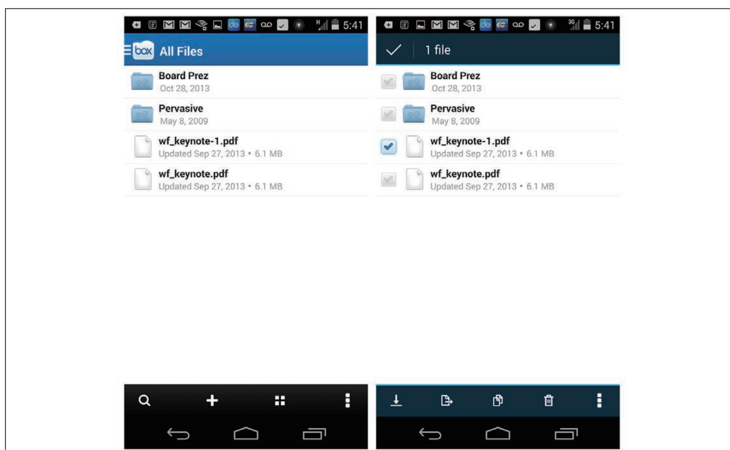



图 5-37

安卓版 Box：长按一个文件可选择该文件，同时显示上下文工具；也可以选择其他条目

# 批量操作

常见的批量操作包括选择、删除和重新排序。为了避免这些选项全部列在每一个条目上而造成主页面臃肿不堪，就要提供批量操作的模式。

iOS 原生应用 Photos 为从相机胶卷中选择照片进行分享提供了一个“选择”模式。当一张照片被选中时，上下文工具（分享和删除）就会显示出来，同时还可以选择更多照片。

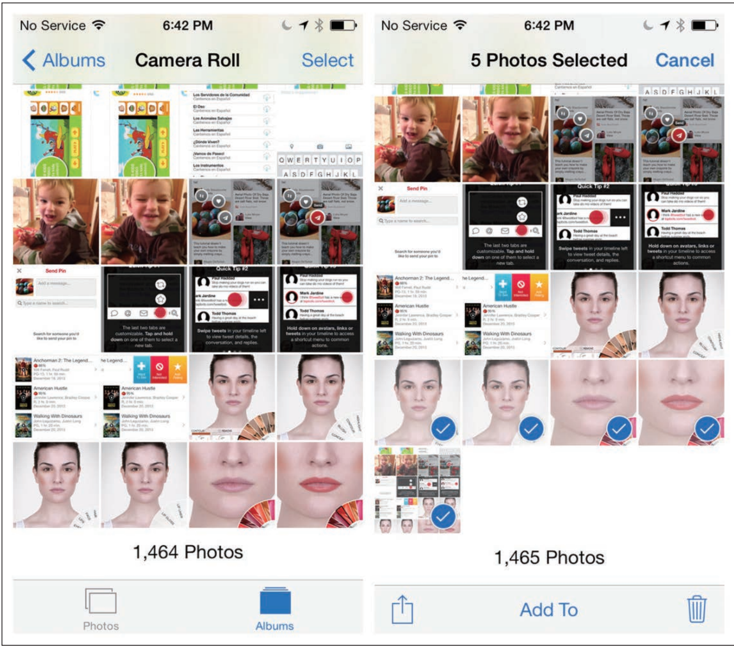
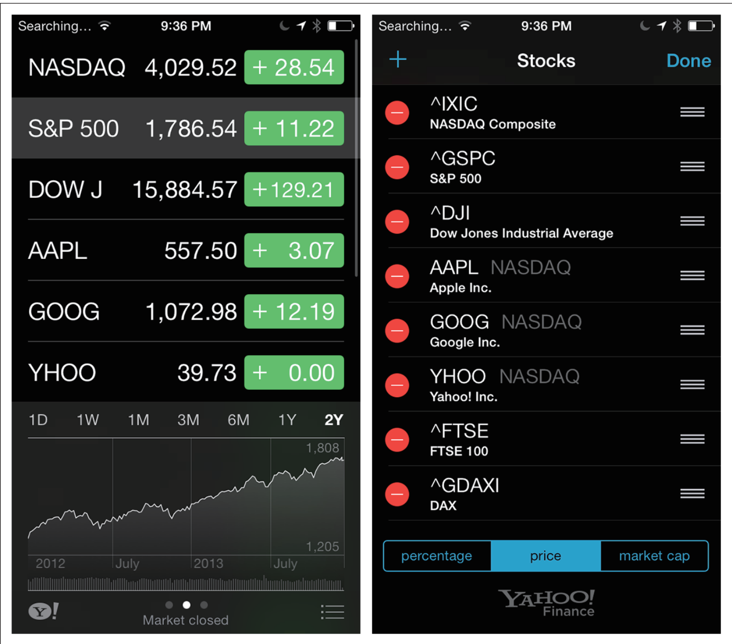



图 5-38
iOS 原生应用 Photos：上下文工具可以对全部选中的条目执行批量操作

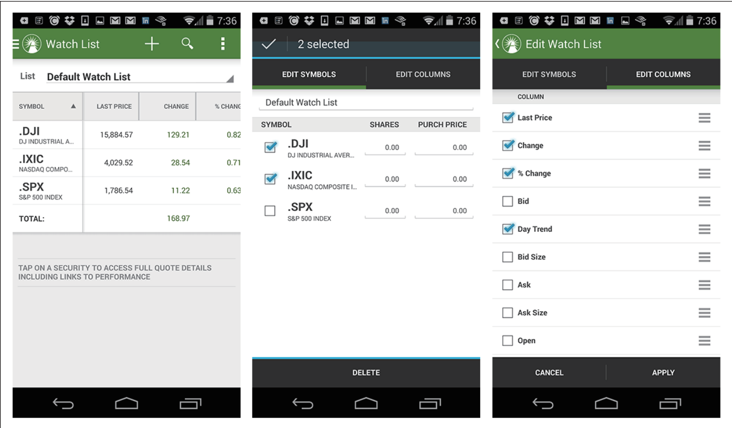
iOS 的 Stocks 应用提供了一种可重排、删除和排序股票的编辑模式。根据 iOS 的设计标准，点击右上角的“完成”会保存变更并退出编辑模式。

图 5-39
iOS 版 Stocks：编辑模式下的批量操作




安卓版 Fidelity 采用了同样的模式，一种是选择股票，另一种是在“关注列表”中对列进行选择和排序。

图 5-40
安卓版 Fidelity：批量选择股票、设置“关注列表”的条件




iOS 版 Path 也在编辑模式中提供了批量操作，如添加、删除和把联系人设为重要联系人（用星形图标表示）。

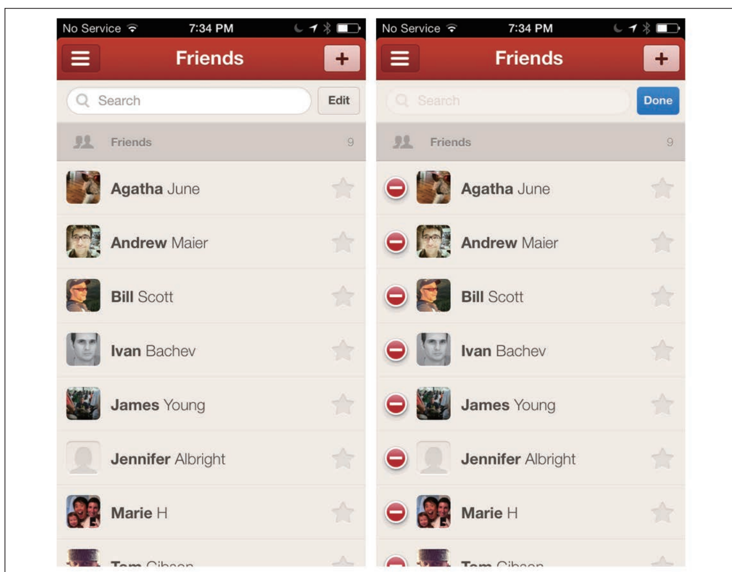



图 5-41
iOS 版 Path：编辑模式中的批量操作可以一次性对多个条目进行操作

在安卓版 Path 中，为了触发编辑模式，你可以在列表中长按任何好友。Path 还可以根据安卓设计原则改进其 UI 设计，显示出选中的好友数量，而不仅仅是把选中项用深色表示。

[小贴士]

像删除和重排这样的批量操作最好使用独立的编辑模式。要提供一个退出编辑模式的明显选项。

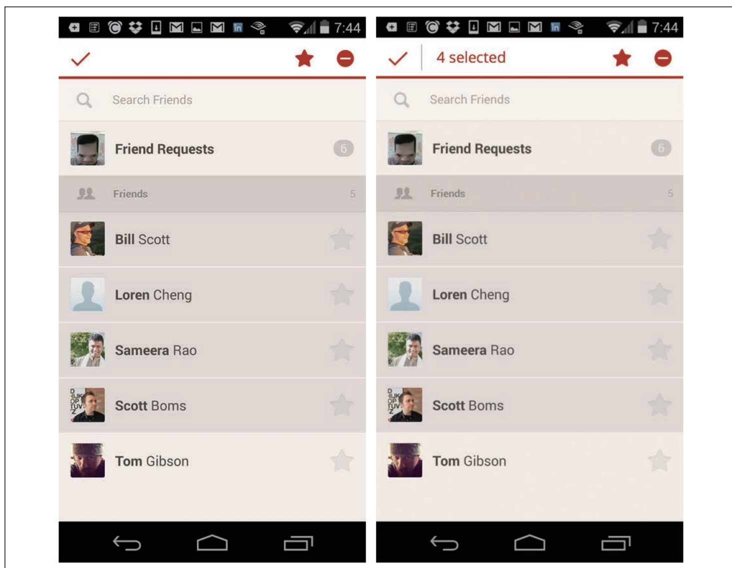



图 5-42
安卓版 Path：左图是现在的设计；右图是建议的改进，显示出选中条目的个数

## 锁屏控件

iOS 和安卓系统都提供了在锁屏界面显示工具的选项。这种模式常见于音乐类应用，但并不限于此。如果你的应用要在后台运行，而用户可能需要快捷方式进行暂停、停止或调整应用，那么可以考虑为锁屏设计工具。

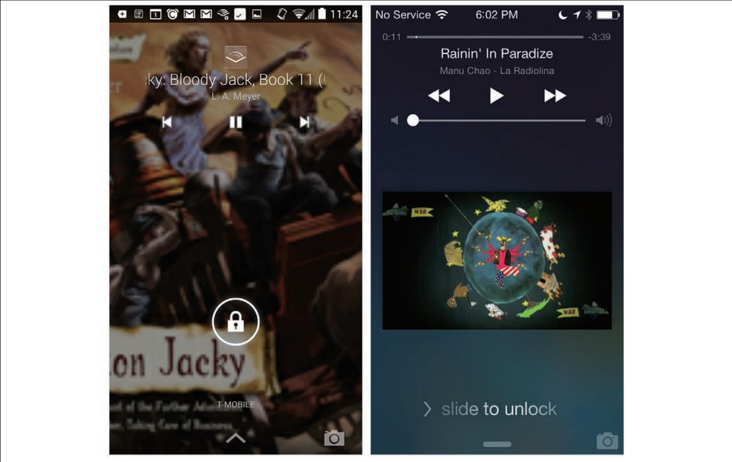



图 5-43
安卓版 Audible 和 iOS 系统的 iTunes：锁屏控件

控件不必限于播放和暂停音视频。安卓版 Any.do 使用了小部件来在早上提醒用户使用，也在一天当中帮助用户管理任务。

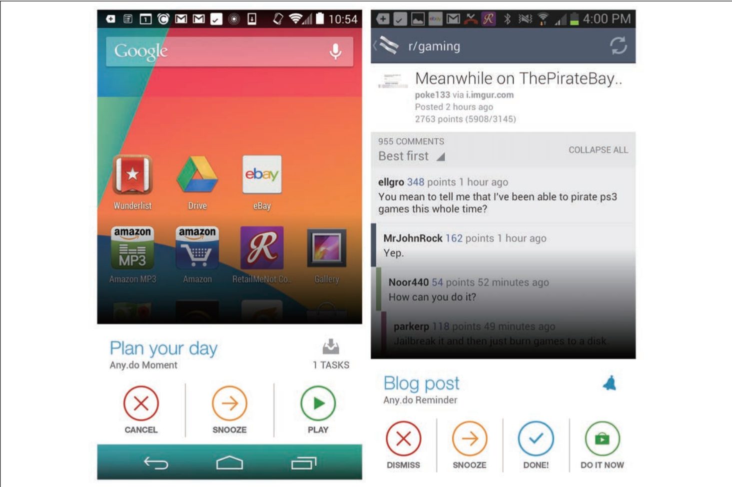



图 5-44
安卓版 Any.do：小部件能够提供比通知更深层的体验

[小贴士]

移动用户体验可以突破应用的常规边界。为提高用户体验，可以考虑采用锁屏控件或小部件。

# 第 6 章

# 图表




- 图表模式

  带有筛选器的图表（Chart with Filters）、交互式时间线（Interactive Timeline）、数据点详情（Data Point Details）、逐级深入（Drill Down）、概览加数据（Overview plus Data）、交互式预览（Interactive Preview）、仪表盘（Dashboard）、缩放（Zoom）、迷你图（Sparklines）、嵌入式图例（Integrated Legend）、阈值（Thresholds）、透视表格（Pivot Table）

为移动设备设计图表是一大挑战，却也为我们提供了机会去接近数据可视化最佳实践的核心。图表应该通过描绘数据之间的重要关系讲述故事。如果没能实现这一目的，那它就是在浪费空间。

在本章第一部分，我会提供一些技巧来帮助你确保为用户设计出清晰而有意义的图表。为了说明这些技巧，我们会介绍一些好的和坏的设计示例。在本章第二部分，我们会讨论常见的图表设计模式。最后，我们将展示两个集大成于一身的应用典范。

## 技巧1：了解基础知识

许多印刷和桌面端的图表设计准则都适用于移动应用的图表设计，其最佳实践也值得借鉴。黄慧敏（Donna Wong）的《< 华尔街日报 > 经济信息图标指南》（*The Wall Street Journal Guide to Information Graphics*，Norton 出版社，2013 年出版）一书对此做出了很好的讲解。SlideShare 网站上有一组演讲幻灯片，是由商业信息分析师 Jen Underwood 制作的“数据可视化最佳实践”（http://www.slideshare.net/idigdata/data-visualization-best-practices-2013），可以当作快速学习图表基础知识的捷径。

用来展示数据的图表样式有许多。你可能对其中一些颇为熟悉，比如柱状图、折线图和饼图。在哪种图表最好这个问题上，专家们多有争论。比如说，Jen Underwood 等许多专家就强烈反对使用饼图，而另一些人则大力支持饼图。我不会偏向任何一方。我见过表达效果很好的饼图（本章中就介绍了一些），也见过设计差劲、不知所云的柱状图。

如果你去 Dribbble 或 Behance 这样的设计图库网站看看，会看到许多美轮美奂的移动端图表。遗憾的是，它们当中许多都是设计习作，对信息的交流并不是特别有用。下面两个示例看上去不错，但都遗漏了用来说明数据背景信息的图表标题。设计师使用的颜色无意间臆造了原本不存在的数据间的关系。




图 6-1

虽然 Dribbble 上的这两个图表设计得很漂亮，但都没能提供有效的信息，让人困惑

关键是要了解基础知识，这样你就可以根据工作需要选用正确的图表，图表包含且仅包含所有必要的信息。必要信息至少应该包括图表标题、带有明确标签的坐标轴，以及数据。

## 技巧2：保持简单

图表越简单越好。注意不要画蛇添足，让多余的数据或视觉效果降低图表的有效性。还记得我之前提到过的专家吗？他们把这种过度装饰叫作“图表垃圾”（chart junk），对他们来说，没有比找出这样的例子拿来嘲笑更有意思的事了。所以为自己好，也为用户好，不要做这样的事情。要了解更多关于图表垃圾的示例，请参看第 11 章“图表垃圾”节。

Argus 应用使用了基本的柱状图和折线图展现某一用户完成步行和跑步目标的进度。

图 6-2

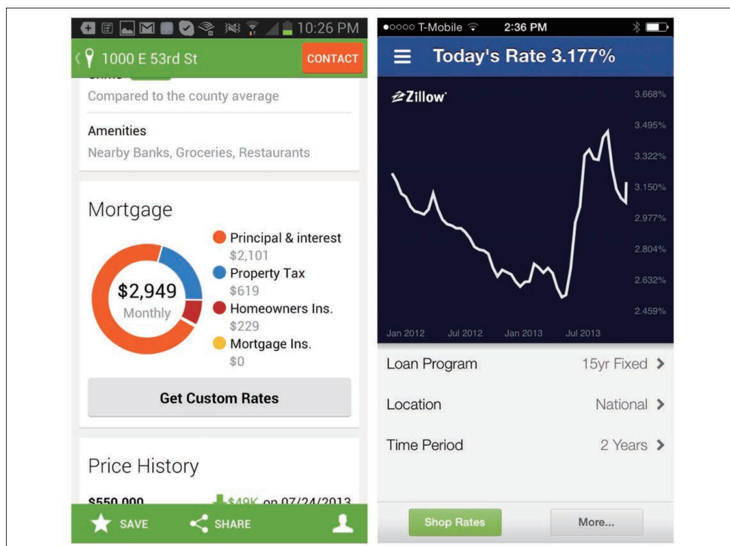
iOS 7 版 Argus：简单的柱状图和折线图展现用户完成健身目标的进度




其他简单、只读的图表在 Trulia 应用中也可见到，这个应用使用了环形图（饼图的另一种形式）来显示一项抵押贷款的分类明细，而 Zillow Mortgage Calculator 应用则使用折线图显示抵押贷款利率的趋势。

图 6-3

简化原则：安卓版 Trulia 中的环形图和 iOS 版 Zillow Mortgage Calculator 中的折线图



## 技巧3：进行用户测试

好了，你的图表看上去非常性感，在你的团队里也是人见人爱。但现在的问题是，你的用户能看懂它吗？为了找出答案，在你进行用户测试时，要像设计数学测验那样设计可用性问题。要向参与者提问关于数据的具体问题。如果他们回答不出来，就说明要么图表形式不对，要么图表太复杂，要么遗漏了关键的文本标签——或者三者皆有。

比如说，在安卓版 1Weather 应用的图表测试中，我可能会向受试者询问以下问题：

- 这张图表显示的是什么时间段？
- 周五下雨的概率是多少？周一又是多少？
- 本周哪一天温度会最高？哪一天最冷？

在测试的过程中，可能就会明确一点：参与者不确定实线和虚线之间的区别。只需增加一个文本标签就可以解决这个问题，但要再做一次测试进行确认。测试期间，我喜欢在参与者回答问题时画草图（或者请参与者画草图），并且不断对图表设计进行修改，直到参与者能够正确回答所有问题为止。




图 6-4
安卓版 1Weather 应用：左图遗漏了图例，右图增加了图例

### 技巧4：了解你的工具集

别急于打开 Photoshop，你应该先熟悉一下你的开发团队正在使用的图表库。有许多不同的图表库可供选择，以下是一些值得探索的热门图表库：

- Core Plot（https://code.google.com/p/core-plot/）适用于 iOS
- ShinobiControls（http://www.shinobicontrols.com/）适用于 iOS 和安卓
- TeeChart（https://components.xamarin.com/view/TeeChart）适用于 Xamarin
- Telerik RadControls（http://www.telerik.com/products/windows-phone.aspx）适用于 Windows Phone
- JS 图表库，比如 AnyChart（http://www.anychart.com/products/anychart/gallery/）或者 amCharts（http://www.amcharts.com/）

让开发人员给你展示这些图表库的可用功能，并就图表设计向他们咨询。除了用户测试之外，设计师和开发人员之间的早期协作是保证图表满足用户需求的最好方式。

好了，你已经学到了一些基础技巧，现在让我们来看看移动应用图表中最常见的设计模式。

## 带有筛选器的图表

时间控件或其他筛选器可以增强一个简单的图表。在 Fidelity 和 StockPlus 这样的应用中，用户可以查看不同时间段的数据。

在采用单页筛选设计模式的时候，要为触摸对象保留足够的空间。在 Chaikin 应用中，筛选控件放在了图表内，人们几乎看不见它们。在本书第 1 版出版之后，Chaikin 重新设计了这个页面，把时间筛选器放到了左边，但这样还是没有解决问题。

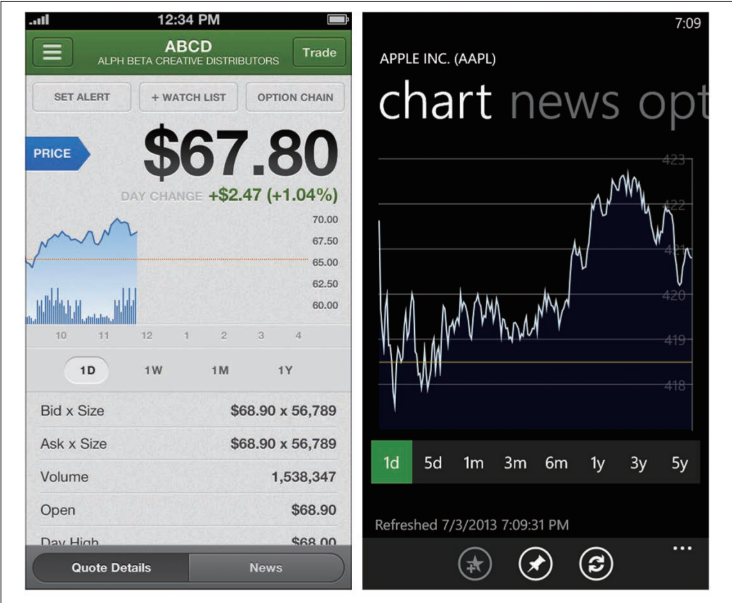



图 6-5

iOS 版 Fidelity 和 Windows Phone 版 StockPlus：按时间段筛选



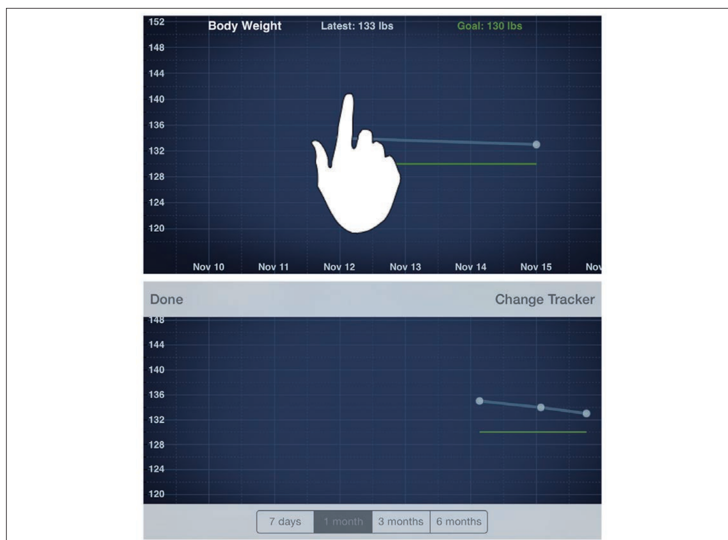

图 6-6

iOS 上两个版本的 Chaikin：筛选控件还是很难看见

DailyBurn Tracker 应用采用了上下文筛选模式，点击图表会在顶端显示操作，在底端显示时间筛选器。

图 6-7

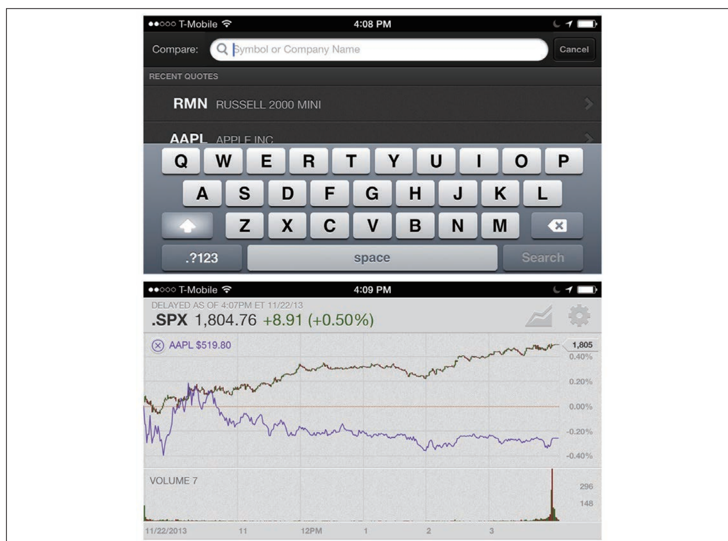
iOS 版 DailyBurn Tracker：点击图标显示筛选器




当然，时间并不是唯一的筛选选项。比如，在 iOS 版 Fidelity 中，可以选择数据来源进行对比。

图 6-8

iOS 版 Fidelity：数据来源对比




**[小贴士]**

使用常见的筛选模式，比如单页筛选、筛选抽屉以及筛选表单（参见第 4 章“筛选模式”一节）。

Fidelity 应用还提供了改变图表类型、阈值和定价坐标轴的设置选项。

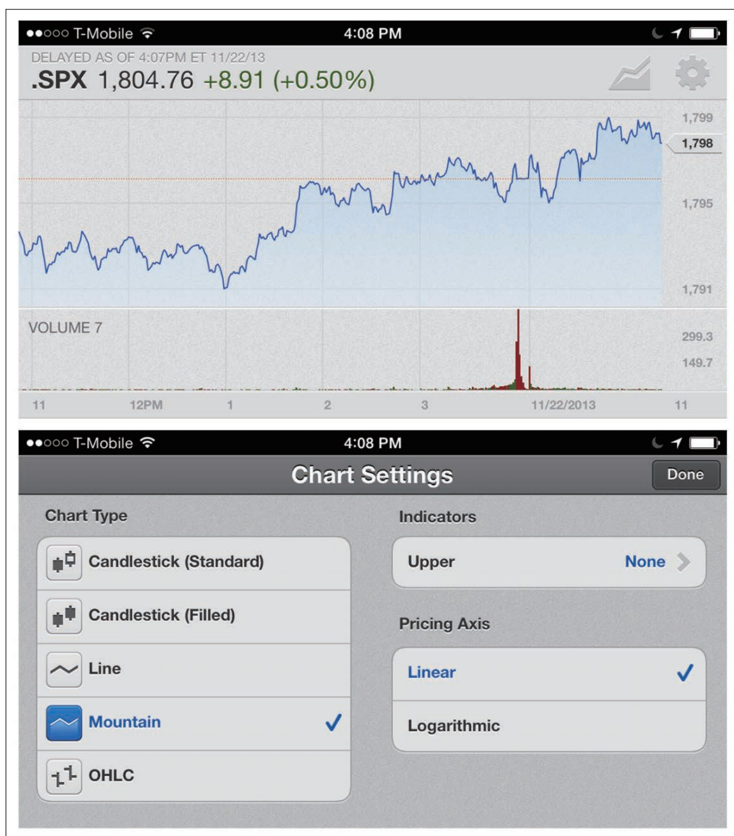



图 6-9

iOS 版 Fidelity：对比图表和图表设置

# 交互式时间线

在本书第 1 版中，我谈到了一种叫作“滚动窗口”（Scrolling Window）的模式，并用 iOS 版 Roambi 作为示例。现在我已经把这种模式扩展至任何与图表或数据可视化结合起来的交互式时间线。

以 Roambi 为例，屏幕底端的小图表提供了一个用来选择时间点的交互式控件（两个拖拽手柄），在上方显示所选时间段内的大图表。

图 6-10

iOS 版 Roambi：用来选择时间窗口的手柄




［小贴士］

比起静态图表或可视化数据，交互式时间线能为用户在选择上下文方面提供更多自由和控制能力。

天气应用 Dark Sky 提供了一个时间线和一个播放按钮，来显示天气随时间变化的动画。它还增加了一个“Now”选项，用来跳回到当前的时间点。

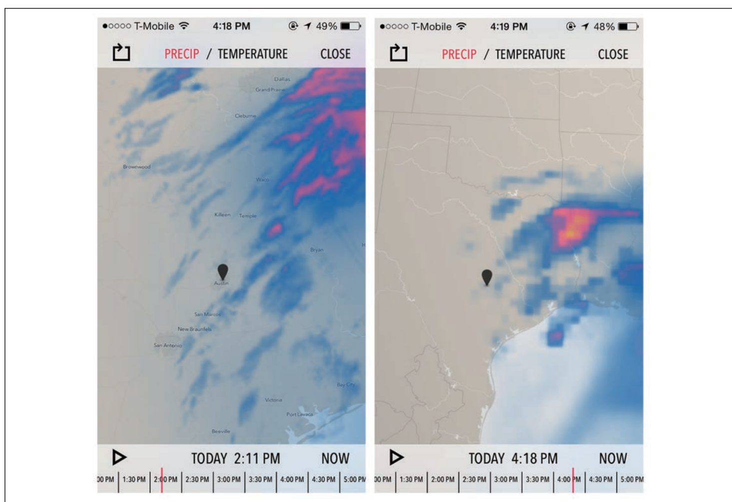



图 6-11
iOS 版 Dark Sky：交互式时间线加播放按钮（http://www.youtube.com/watch?v=d_ZJYVPxRe8&feature=youtu.be）

## 数据点详情

因为悬浮设计在移动端行不通，而长按又不是特别直观，所以就需要一种有别于前两者的交互来显示特定数据点的详情。在 iOS 版 Zillow Mortgage Calculator 中，点击饼图中的一个扇区可以显示款项的分类详情。

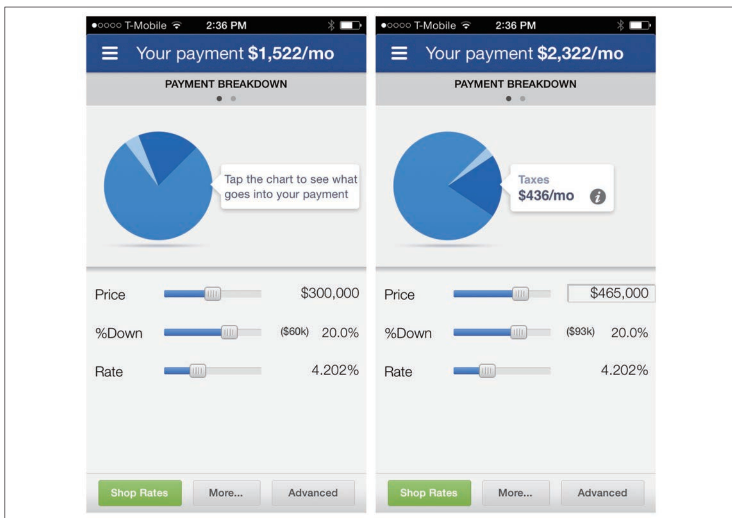



图 6-12
iOS 版 Zillow Mortgage Calculator：点击图表区域显示数据点详情

在 Apple Stocks 和 Yahoo! Finance 应用中，点击一个数据点，它就会在图表底端的预览窗口（Preview Window）中高亮显示出来（虽然你得尽力靠近才能看到）。在这些应用中，整个图表就是一个触摸对象，所以你可以点击图表的任何位置，然后拖拽调整到特定的日期。




图 6-13

iOS 版 Apple Stocks 和 Yahoo! Finance：点击数据点查看更多信息

在 Apple Stocks 应用中，触摸一点会显示该数据点详情，但同时触摸两点（多点触摸）会显示两个数据点的差异。




图 6-14

iOS 的 Apple Stocks：单点触摸和多点触摸时的数据点详情

在 Traffic for Google Analytics 中，数据点很大，你完全可以点击它们，但这个应用也会自动高亮你滑动到的月份数据。在我向前滚动时间时，首先高亮显示 9 月，然后是 8 月，最后滚动回到 5 月，如此这般。

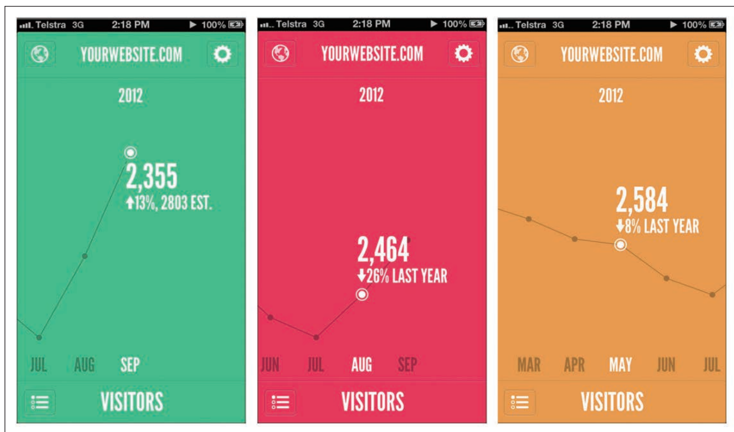



图 6-15
iOS 版 Traffic for Google Analytics：你可以使用手势查看数据点详情，并改变时间段（http://www.youtube.com/watch?v=UqZkxNT3Ggs）

iOS 7 版的 Roambi 采用了一种巧妙的设计，你只需点击拖拽手柄就可以切换多个数据来源，查看其详情。

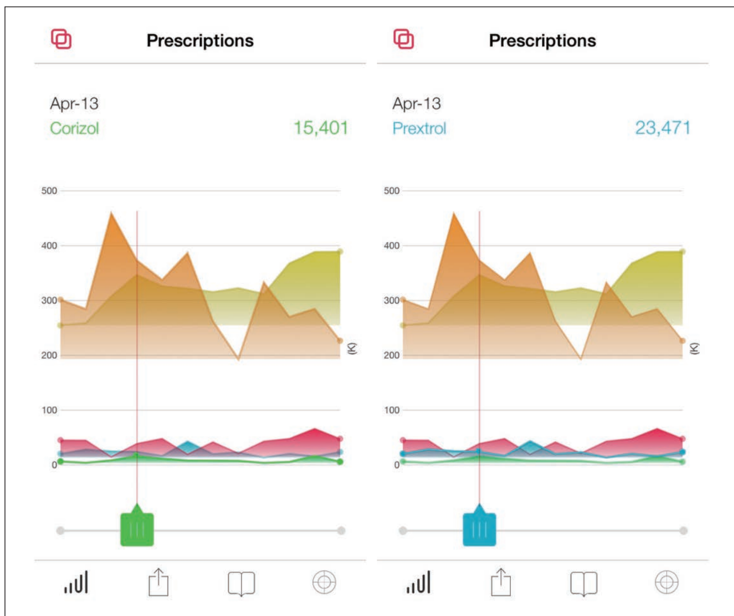



图 6-16
iOS 版 Roambi：数据点详情切换器（http://www.youtube.com/watch?v=XBipN_7XMnU）

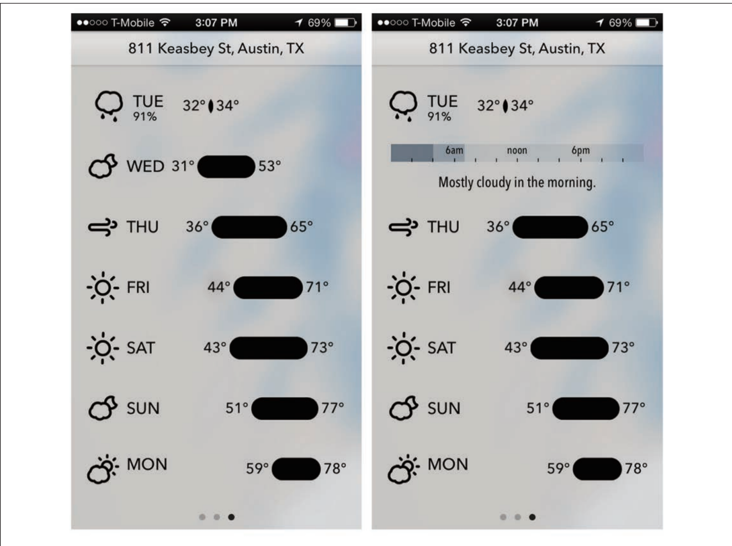
Dark Sky 鼓励用户通过滑动查看详情。当载入周视图时，第一行会弹跳出来提示你使用滑动手势。滑动该行后，当天的天气详情就会显示出来。

［小贴士］

我们的网页体验已经给用户设定了预期效果——数据点详情会通过悬浮显示，但这在移动应用上行不通。可以考虑在用户点击数据点获取更多信息时再提供详情。

图 6-17

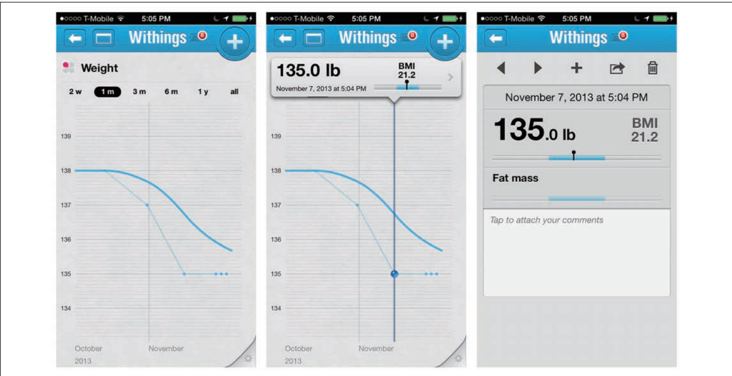
iOS 版 Dark Sky：滑动行显示详情（http://www.youtube.com/watch?v=d_ZJYVPxRe8）




Withings 把数据点详情和逐级深入模式结合了起来，我们会在后面进一步介绍。点击数据点会显示详情，然后点击详情浮层会进入新页面，你可以在这个页面进行编辑或其他操作。

图 6-18

iOS 版 Withings：点击图表显示数据点详情；点击详情跳转到有更多详情的新页面




［小贴士］

吸引用户逐级深入查看更多数据。使用系统相应的控件来提供返回路径（即 iOS 系统的“返回”按钮和安卓系统的“向上”按钮）。

## 逐级深入

如果想在上下文中多显示一点信息，通常采用数据点详情模式就足够了。但有时候，用户需要更深入地查看数据。这时，逐级深入模式就会非常有用。

以 Mint 为例，顶级视图是仪表盘式，下一层是预算列表页，再下一层是有更多详情数据的单个预算页（以及编辑阈值的按钮）。

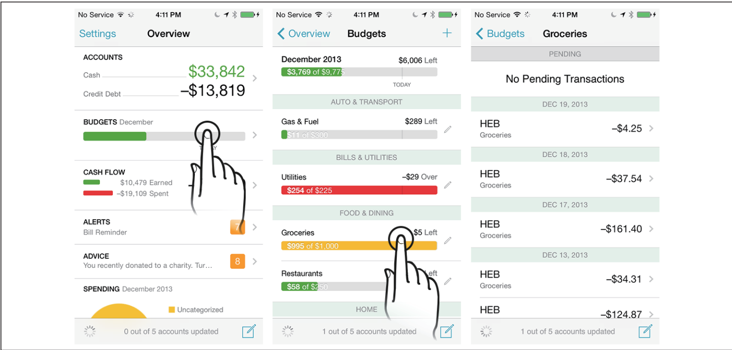



图 6-19

iOS 版 Mint：逐级深入显示详情层次

如果没有清晰的视觉提示可以点击进入下一层，那么你应该使用导引，鼓励新用户逐层点击进入下一页面（参见第 8 章）。

# 概览加数据

我们在第 3 章中也谈到了概览加数据模式，但在这一章里，它专指使用图表显示的概览。这种图表模式是过去几年中变化最大的模式之一。

本书第 1 版主要讨论了在上方显示趋势图表、下方显示数据详情表的设计模式，像 Gaug.es 和 Personal Capital 设计的那样。

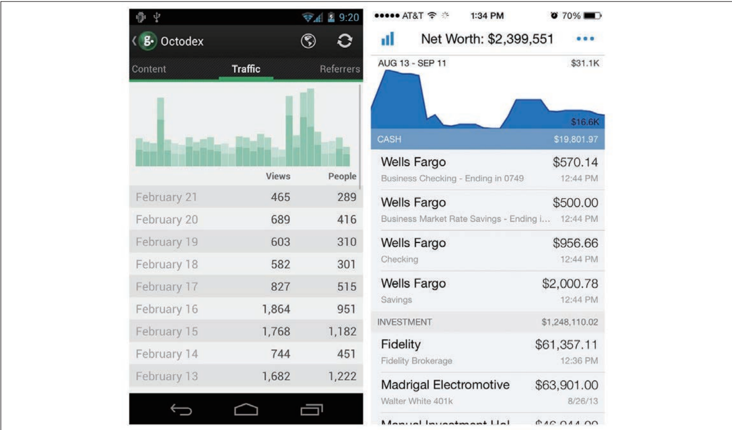



图 6-20

安卓版 Gaug.es 和 iOS 版 Personal Capital：趋势图表位于上方，详细数据表位于下方

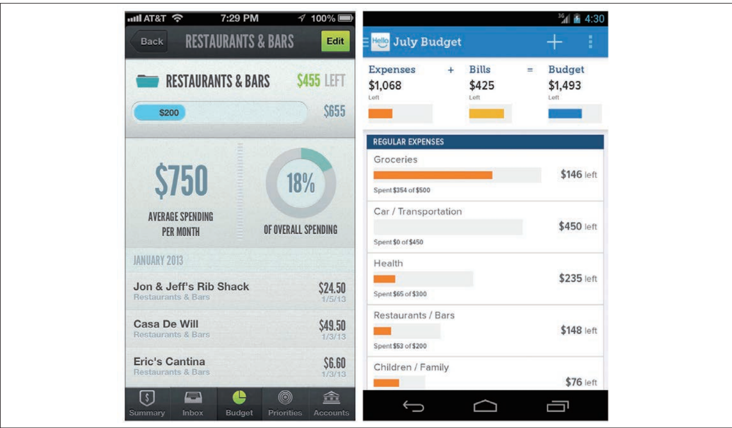



图 6-21
iOS 版 LearnVest 和安卓版 Hello-Wallet

但这种模式并不只限于只读图表和其各自的数据表格。比如在 Personal Capital 应用中，交互式环形图表用来显示概览。点击一个扇区，下方的表格就会突出显示相应的类别和总数。

这个应用的一个早期版本还在环形图表中间放置了一个迷你图（一种很小的图表，在本章后面会做介绍），为选中类别的支出显示了更多的信息。这种设计后来被新的设计所取代，在新的设计中有显示增减百分比的数字，并在圆环后面的背景中放置了随时间变化的区域图。

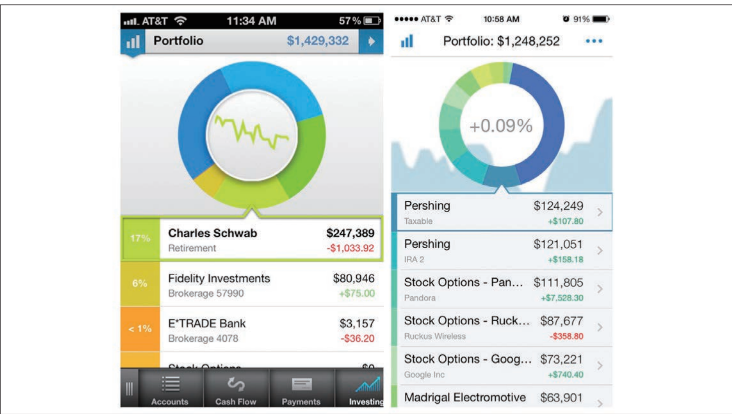



图 6-22
iOS 版 Personal Capital 的两个迭代版本：迷你图被背景区域图取代

**［小贴士］**

对用户来说，带有详细数据的可视化概览比没有摘要的数据列表更有价值，在这种情况下可以采用这种模式。

概览加数据模式的另一个变体可以在安卓版和 iOS 版的 Move 应用中看到。概览用气泡图呈现，而详细数据则用时间线显示在下方。

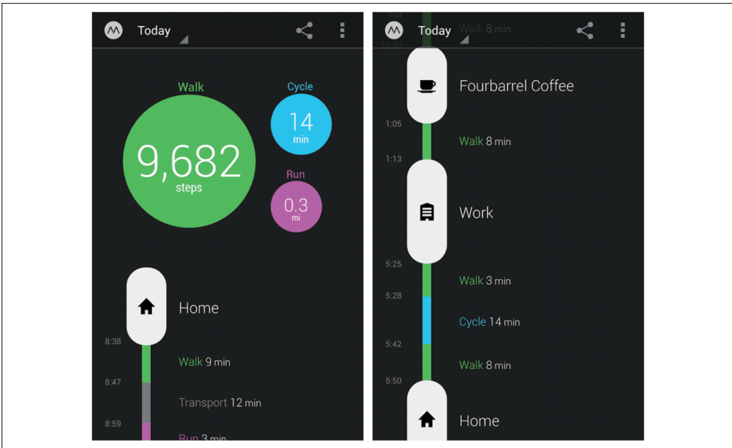



图 6-23
安卓版 Move：概览加数据的另一种设计方案

## 交互式预览

这种模式在第 2 章中也曾提到。在 Trulia Mortgage Calculator 的例子中，你可以看到，通过滑动页面下半部分的滑块，我的年收入从 7.5 万美元增加到了 29.5 万美元，页面上半部分的饼图也随之重绘，同时图例也更新了。

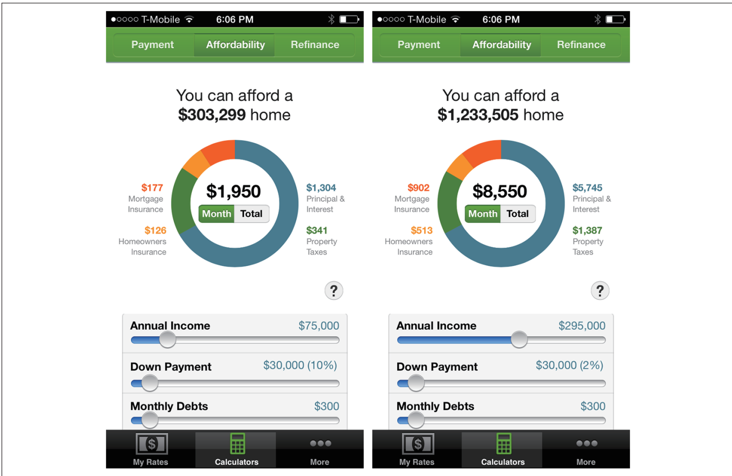
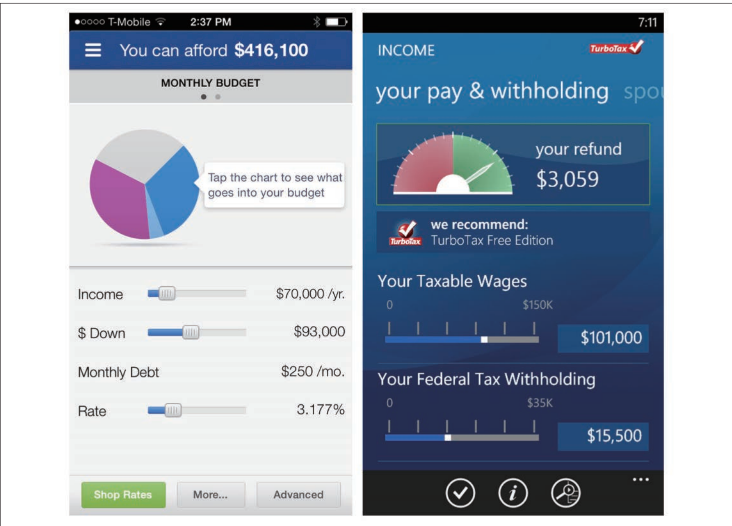



图 6-24
iOS 版 Trulia Mortgage Calculator：交互式预览模式的图表

Zillow Mortgage Calculator 和 Intuit's TaxCaster 运用了同样的交互设计。

图 6-25

iOS 版 Zillow Mortgage Calculator 和 Windows Phone 版 TaxCaster：交互式预览




［小贴士］

交互式预览模式适用于需要显示多个预测的场景，常见于个人理财类的应用。

Mint 在预算页面采用了交互式预览的设计。这个页面的初始状态显示了选中类别下过去 6 个月的支出折线图。这个页面还设计了一个视觉吸引（见第 8 章），引导用户“点击或拖拽以改变预算”。当用户调整手柄时，新的预算预览图就会显示在页面上方，直到松开手柄为止。

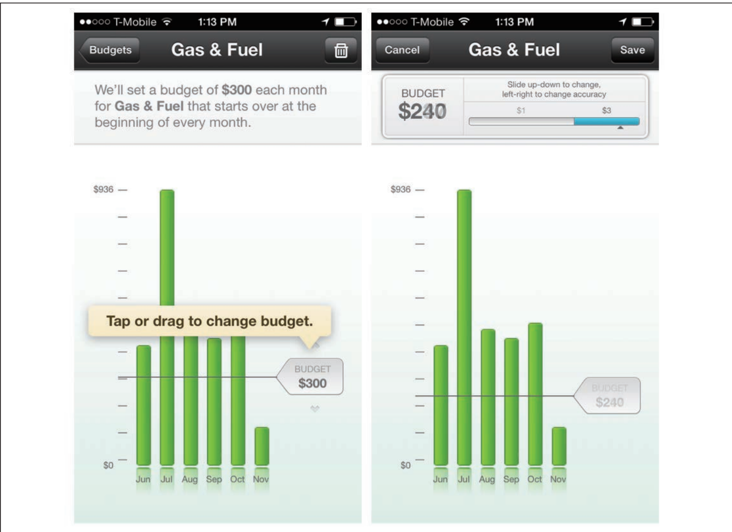



图 6-26

iOS 版 Mint：设置预算的交互式图表

# 仪表盘

这种模式在第 1 章中也有所涉及。仪表盘可以用作主导航，比如 Withings 和 Mint，它们都采用了仪表盘作为进入指定模块的导航中枢。

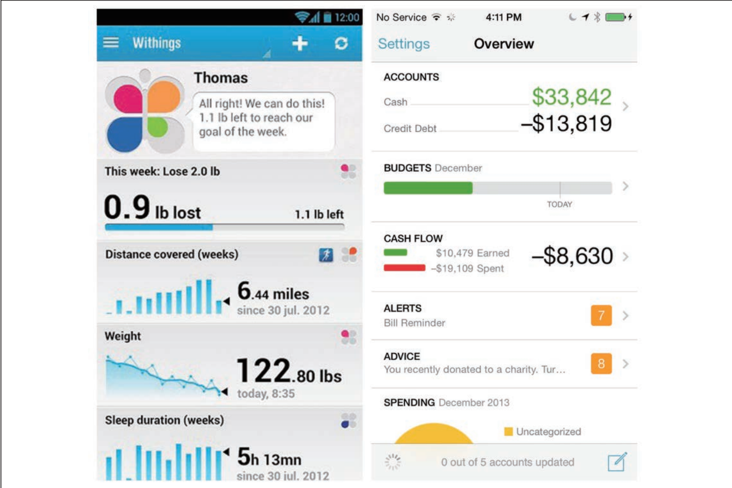



图 6-27
安卓版 Withings 和 iOS 版 Mint：仪表盘用作主导航

Gekoboard 使用了一种可自定义的仪表盘来跟踪销售数据。你可以添加、删除或重新排列其中的小部件。

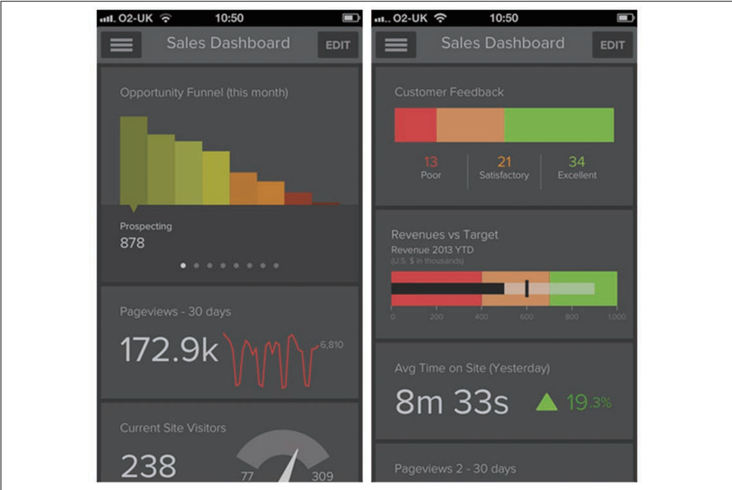



图 6-28
iOS 版 Gekoboard：可自定义的仪表盘

［小贴士］

如果同时显示多个图表可以更好地说明一个多维度信息，那么可以采用仪表板模式。如果只是合并不同的无关联的数据，就不要使用这种模式。

仪表盘也可以用作次级导航，比如 iOS 版的 RunKeeper 和安卓版 Lose It!，二者都使用了多个叠加图来显示相对于一组目标的活动进度。

Analytiks 采用了仪表盘同时作为主导航和次级导航。首页显示网站的关键绩效指标，你可以点击进入次级仪表盘查看关于用户特征、浏览器、受众和访问持续时间的详细指标信息。

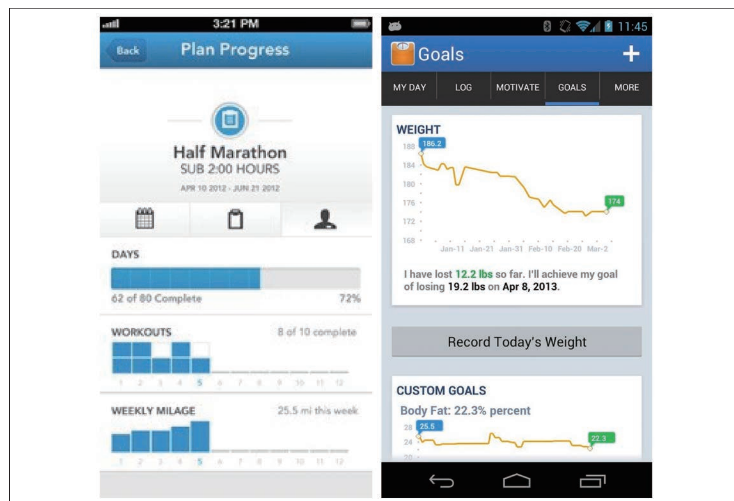



图 6-29

iOS 版 RunKeeper 和安卓版 Lose It!：仪表盘作为次级导航




图 6-30

iOS 版 Analytiks：设计精美的仪表盘同时用作主导航和次级导航

## 缩放

如果页面上有很多元素，图表只是其中一个，用户就会很难看清图表。常见的做法是引导用户旋转至横屏，查看全屏图表。一般会隐藏标题和导航元素，尽可能为图表的展示留出足够的空间。旋转屏幕至竖屏时，导航和其他控件恢复原状。

为旋转屏幕查看浸入式图表视图提供一个视觉吸引，这一点很重要。NASDAQ QMX 应用设计了一个旋转手机的小图标，而 SigFig 则在点击图表时显示旋转屏幕的文字提示。

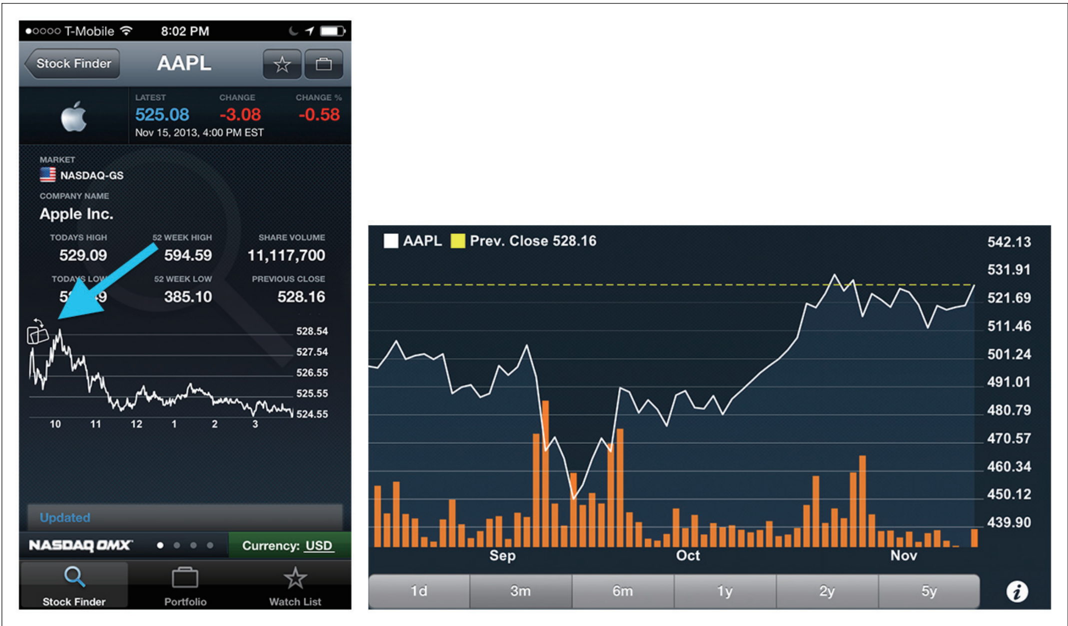
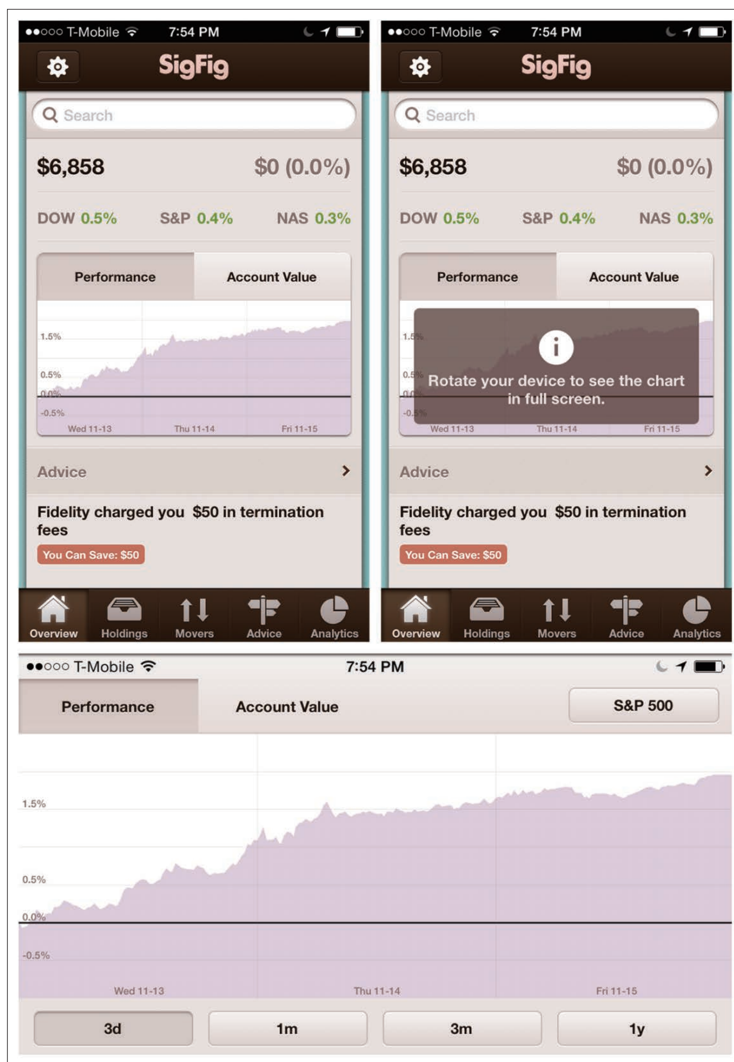



图 6-31　iOS 版 NASDAQ QMX：图标引导用户旋转屏幕查看详细视图

图 6-32

iOS 版 SigFig：点击图表弹出旋转屏幕查看详细视图的文字提示




[小贴士]

要吸引用户旋转屏幕至横屏，以查看全屏的详细图表视图；在设备旋转至竖屏时自动恢复导航。尝试用缩放或平移的手势来扩展或缩小图表的焦点。

当图表处于横屏模式时，用户可能仍然想更仔细地查看特定时间范围内的图表详情。在雅虎（Yahoo!）应用中，默认视图的时间段为两周，然后放大到一周，再放大到两天，最后可以放大到一天当中的下午 12：30 到 14：20。要再次查看更宽泛的时间段，可以点击时间筛选器或用手势缩放页面。




图 6-33
iOS 版雅虎应用：横屏视图下，缩放可以调整时间范围焦点

# 迷你图

迷你图也称微型图，最初是由可视化信息设计大师 Edward Tufte 提出的，他对迷你图的定义为“一个微小密集、简单明了、单词大小的图表”。

迷你图特别适合移动设备，因为它既提供了数据趋势或变化的概览，又不会占用太多空间。按 Tufte 的话来说，它们可以“嵌入到句子、表格、标题、地图、数据表、图像中”。我认为它们就是很小的图表。

以 iOS 版 Analytics Tiles 为例，上方的两个磁贴使用了迷你图来表示页面浏览量和访问量趋势。可以点击这些磁贴深入到下一级，查看更大、更详细的图表。Gaug.es 也与此类似，在每行都用一个迷你图来表现数据的趋势。

在 Fitbit 的示例中，用子弹式迷你图显示相对预设目标的活动进度。而另一个健康类应用 Noom 则使用微型环形迷你图指示每一餐的均衡程度。

［小贴士］

使用迷你图为信息增加快速直接的视觉冲击，但要通过用户测试验证图表是否明晰。可以用迷你图作为查看详细视图的入口。

图 6-34

iOS 版 Analytics Tiles 和安卓版 Gaug.es：迷你图示例

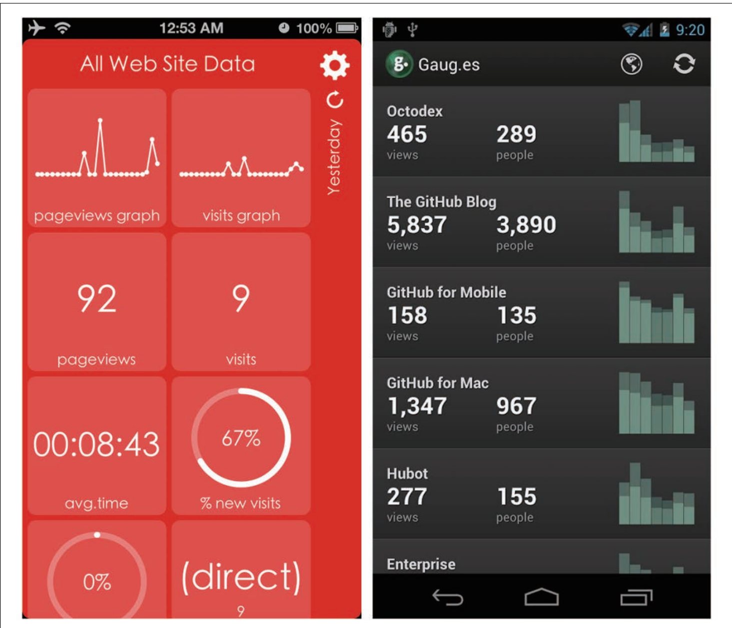
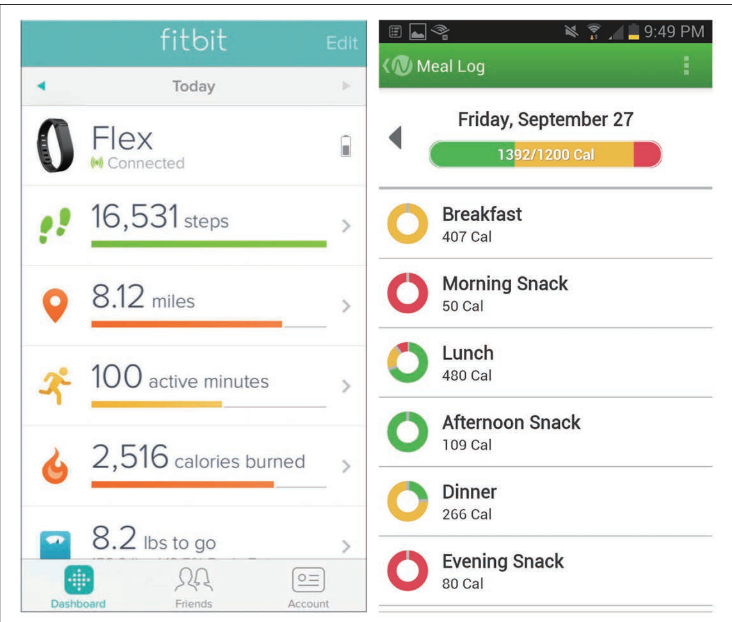



图 6-35

iOS 版 Fitbit 和安卓版 Noom：迷你图的更多示例



## 嵌入式图例

还有一种节省屏幕空间的技巧，是把图例嵌入到图表标题或数据表格下方。iOS 版 Personal Capital 就是一个很好的范例。

[小贴士]

嵌入式图例可以节省宝贵的屏幕空间，但在采用这种模式代替独立图例前要考虑可用性原则。

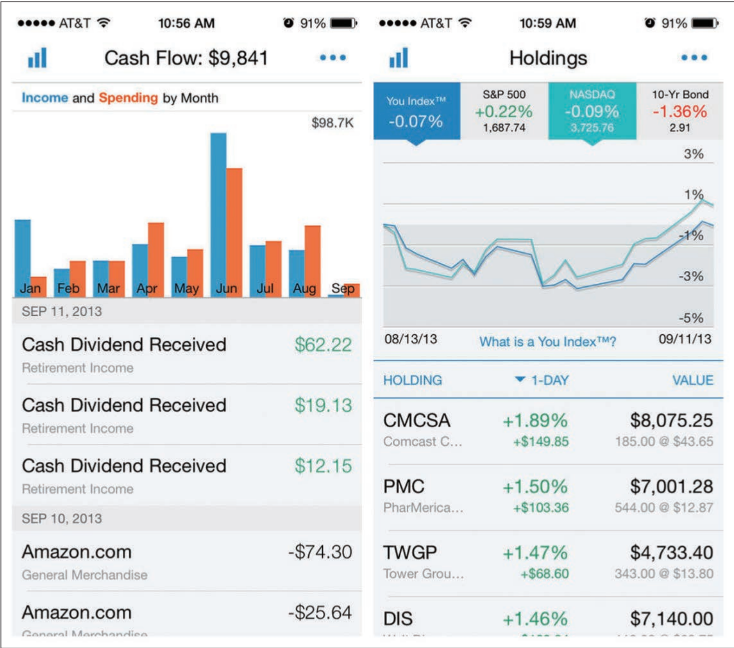



图 6-36

iOS 版 Personal Capital：左图为显示在标题中的嵌入式图例，右图为显示在数据表中的嵌入式图例

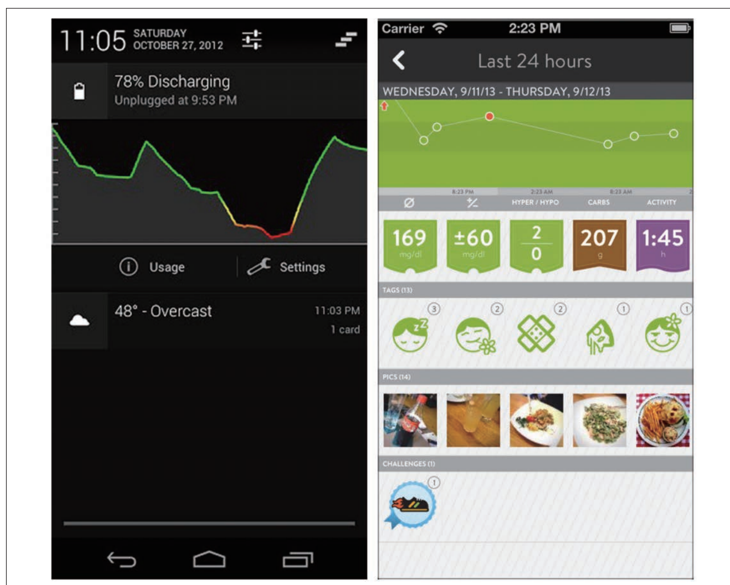
## 阈值

许多图表都使用这种模式表现数据是在范围内还是超出了范围，是正中目标还是偏离了目标。阈值可以用标注在图表上的颜色、条带或线条表示。

安卓版 Battery Graph Notification 和 iOS 版 MySugr Diabetes Companion 应用使用了常见的红 / 黄 / 绿色板表示数据是否可以接受，或是否在目标范围内。

图 6-37

安卓版 Battery Graph Notification 和 iOS 版 MySugr：不同颜色显示数据是否在目标范围内




作为用条带或线条指示阈值的示例，Data Usage 用橙色线条表示警告线，最上方的红色线条显然非常重要；Anthony Lagoon 的监测糖尿病病情的应用用一个条带表示正常范围，任何位于条带上方或下方的数据点都会被认为有问题。

图 6-38

安卓系统的 Data Usage 和 Anthony Lagoon 公司的糖尿病概念应用：分别为阈值线和阈值条带




［小贴士］

阈值对标示个人用户目标特别有用。可以把阈值标记设计成交互式（比如，在图表中可以调节阈值）。

Toshl 和 Lose It! 的子弹图示例中分别使用了虚线和彩色线表示阈值。

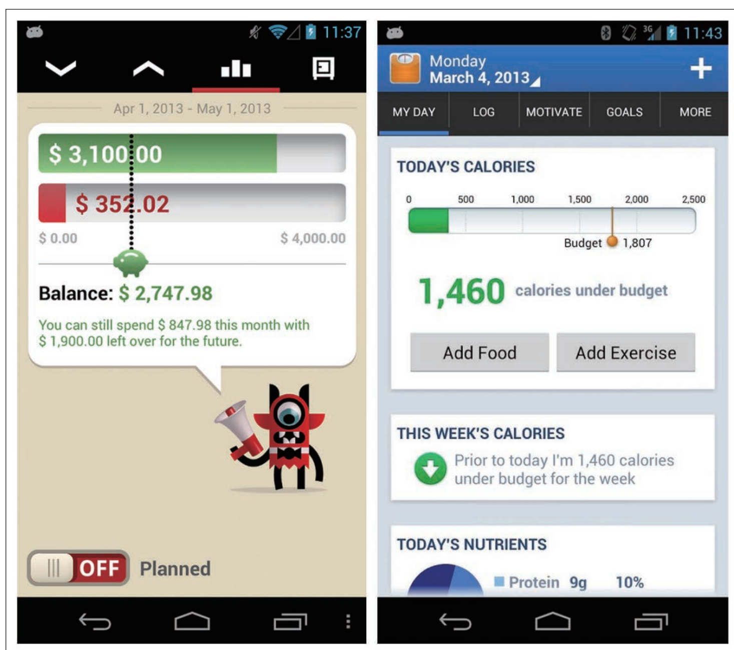



图 6-39
Windows Phone 版 Toshl 和安卓版 Lose It!：更多阈值设计示例

## 透视表格

透视表格是置于大表格当中的小概览表格。举例来说，在流行的桌面和网页电子表格应用中，用户可以通过拖放概览中要显示的数值创建透视表格。用户可以在表格中移动这些数值以多种方式“透视”该数据。

Roambi 开发了一种简单的筛选界面，可以用类似于透视表格的方式查看数据。y 轴保持不动，x 轴可以选择任何维度（选中项会用橙色框高亮显示）。对于每一个维度，可以同时选择特定的类别。下图显示了 Roambi 的简便操作设计：查看所有城市的季度利润，然后快速筛选图表以查看书籍这一特定产品线的相关信息。再增加一项渠道类型的选择，就可以再次缩小数据范围，只显示线上渠道相关的数据。




图 6-40 iOS 6 版的 Roambi：透视表格风格的筛选

## 整合

现在让我们来看一些拥有绝佳图表设计的应用示例。

Nike+ FuelBand 应用以其整洁的数据可视化图表出色地完成了激励用户的任务。其设计师选择了最佳的图表形式，以表示不同目标的进度情况，同时避免了充斥页面的图表垃圾。每一个图表都有清晰的标题，坐标也有清晰的文字标签。这个应用设定了阈值的色板，并且将其运用到每个页面的设计。抛开图表类型不说，进度和表现频谱在数值低的一端总是显示为红色，而高的一端总是显示为绿色。




图 6-41
iOS 版 Nike+ FuelBand：所有页面中清晰干净的数据可视化设计

Weathertron 是另一个数据可视化精心设计的优秀典范。新手引导可以改进为嵌入式视觉吸引（这一点在第 8 章中有相关介绍），使其用户体验简单而直观。比如说，抽屉中紧凑的柱状图可以通过点击右下角显示出来，这种图表形式是显示 7 日天气预报最高温度和最低温度的最佳方式。




图 6-42　iOS 版 Weathertron：设计精美、简单直观的数据可视化

第 7 章

# 引导和视觉吸引




- **引导原则**

  少用文字（Use Less Text）、不要前置引导（No Frontloading）、给予用户成就感（Make It Rewarding）、强化学习（Reinforce Learning）、倾听用户的心声（Listen to Your Users）

- **视觉吸引模式**

  提示（Tips）、新手引导（1st Time Through）、持续的视觉吸引（Persistent Invitations）、可发现的视觉吸引（Discoverable Invitations）

在本书第 1 版中，我谈到了 8 种常见的视觉吸引模式。但现在我们知道，虽然这些模式很常见，但其中没有几个是非常有效的。2012 年，我有幸见到了 Intuit 公司的几位主要设计师。他们告诉我，调查结果显示，这些模式中的大多数都在用户测试中表现不佳。这与我们得出的结果一致。我和我的团队此前也一直在各种应用中测试这些模式，其中包

括 iOS 版 RetailMeNot 和安卓应用。

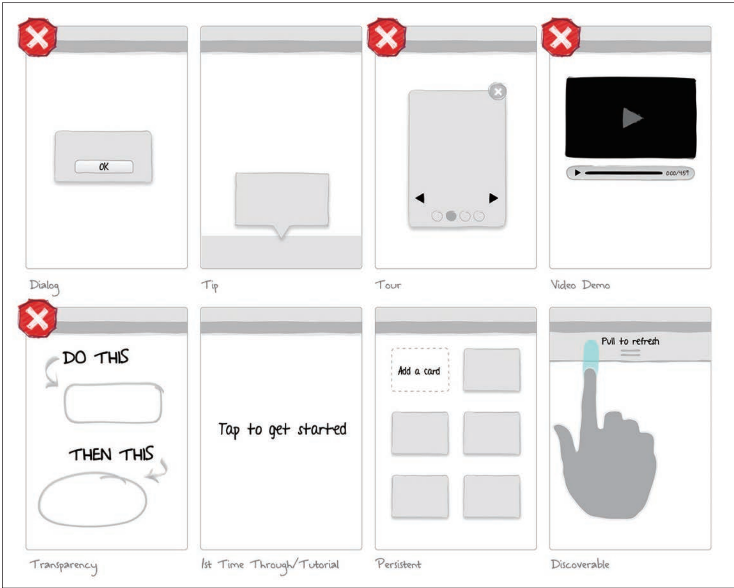



图 7-1
视觉吸引模式的第 1 版

简而言之，测试证明，用户往往会跳过或忽略对话框（Dialogs）、使用向导（Tours）、视频演示（Video Demos）以及幻灯片（Transparencies）。往好了说，用户觉得这些导引不方便；往坏了说，这些导引会让正打算使用应用的新用户十分反感。正如 RetailMeNot 的一位测试参与者所说："我只想立刻进入应用，开始探索。"

所以我们根据这一新的认识撰写了新的一章。本章的前半部分专门讲解如何创造有效的"新手引导"体验，也就是常说的引导页或着陆页。本章后半部分介绍已被验证有效的视觉吸引模式，包括技巧、新手引导、持续的视觉吸引、可发现的视觉吸引。

## 引导原则

那么，为什么对话框、使用向导、幻灯片和视频演示对新用户的引导效果不好呢？我试着向游戏设计领域寻求答案。游戏设计师们向来都懂得，你不能把新玩家直接丢到枪林弹雨之中，指望他们乐在其中。大多数玩家还来不及搞清怎么开火反击就死掉了。

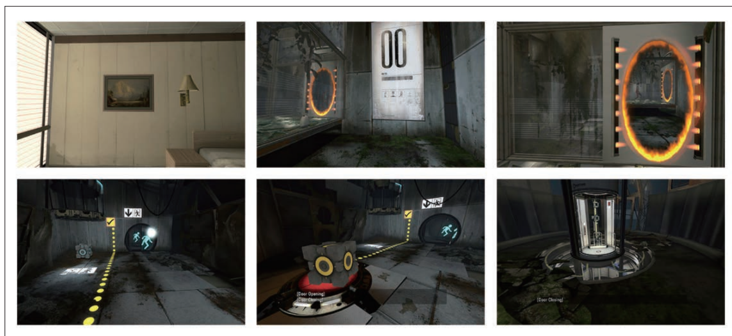


图 7-2
Portal 为新玩家设置了一个安全的环境，让他们在正式进入游戏前熟悉各种操作方法

在游戏设计中，一些推动深度互动的方法要比其他方法奏效得多。在移动应用中，情况也是一样。虽然重要关头不涉及虚拟的生死存亡，但用户在茫然无措的情况下所体验到的挫折感是一样的。如果你的很多用户都有这种挫败的感觉，那么你的应用就完蛋了。

《额外加分》（Extra Credits）在线视频系列从内行的角度深入解析了游戏世界的各个方面。其设计团队制作了一集非常棒的视频，叫作《教程 101》（http://www.youtube.com/watch?v=BCPcn-Q5nKE）。我相信每个应用设计师在看了这部教程之后都会受益匪浅。（忽略视频里那烦人的花栗鼠声似的旁白吧，这部片子值得你这么做。）

《教程 101》罗列出了一些打造高效的游戏引导的基本原则：

- 少用文字
- 不要前置引导
- 让引导更好玩
- 在游戏中强化学习
- 倾听玩家的心声

我们将抛开设计模式，用这些原则作为参照，打造出一个超赞的引导（为适用于应用，我们对这些原则做了少许调整）。随着我们深入探讨这些原则，你就会明白为什么本书第 1 版中的大多数视觉吸引模式没能有效地帮助用户从应用中获得更深层、更丰富的体验。

## 原则1：少用文字

当我们想要解释一件事情的时候，文字通常是最简单的工具。但如果我们想要了解一件事，却经常不愿采用文字这种途径。

《教程 101》指出，我们应该避免只依赖文字进行讲解，因为“阅读文字速度很慢，而且会破坏浸入式体验，此外，那些最需要引导的玩家反而常常跳过文字引导。”

［小贴士］

抑制自己使用文字的冲动，能用图像表现，就不要用文字描述。用文字来指出“因”，然后让用户通过操作观察“果”。

使用太多的文字是一种“说而不秀”的方法，这与移动应用的优势相违背。相反，引导应该是“秀而不说”—应该具有交互性，这样用户可以通过操作来学习。如果我们预先练习过操作，而不仅仅是被告知如何做，那么我们在需要的时候就会更容易记住这些操作。

“少用文字”原则并不意味着用文字永远是错的，或者说，文字总是越少越好。总体而言，它指的是在你想吸引用户进一步体验的时候，应该首先考虑减少文字，增加交互。

下面对比了一些违背了此原则与采用了正确方法的应用。第一组对比的是两个美食应用。Ness 通过文字说明而不是操作演示来介绍定制化页面，这样用户要经过冗长的流程之后才能进入应用。相反，Foodspotting 则让用户直接上手使用应用，其界面上只使用了几个行为召唤操作的提示，用户可以轻松开始体验，不需要文字导引。

还有类似的其他示例，展示了在这方面设计优秀的应用和效果稍逊一筹的应用。

### Ness和Foodspotting

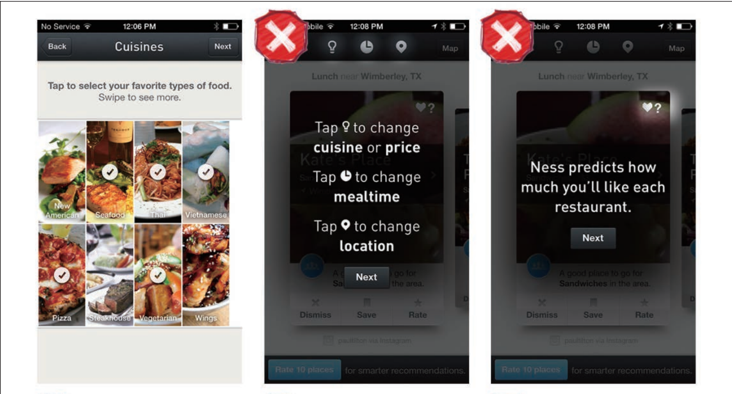



图 7-3

iOS 版 Ness：注意这里在用文字解释操作和效果，而不是鼓励用户操作，让他们自己观察效果

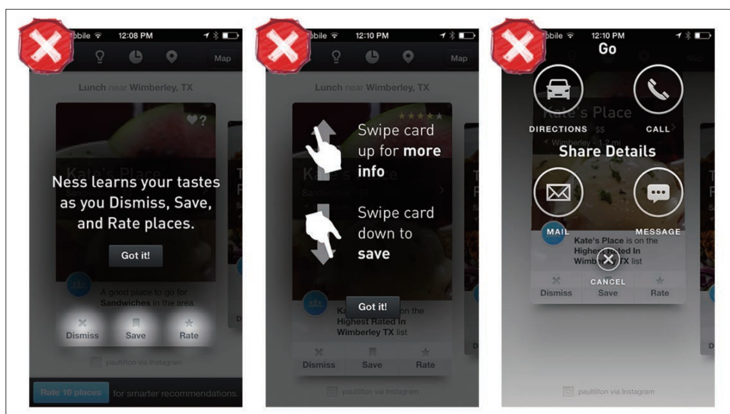



图 7-3（续）
iOS 版 Ness：注意这里在用文字解释操作和效果，而不是鼓励用户操作，让他们自己观察效果

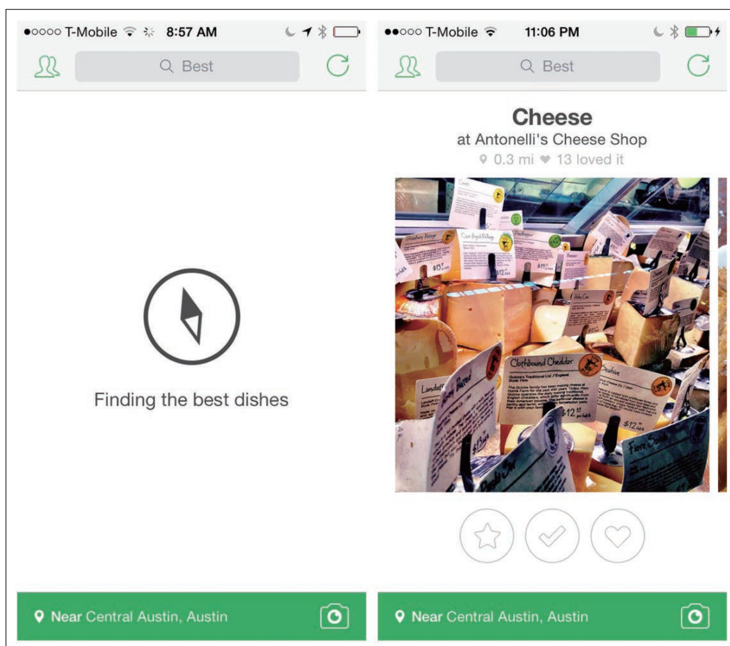



图 7-4
iOS 版 Foodspotting：一个非常简单的应用，让用户直接深度体验，无需引导

## Boomerang和Mailbox




图 7-5
安卓版 Boomerang：文字太多（也见原则 2）

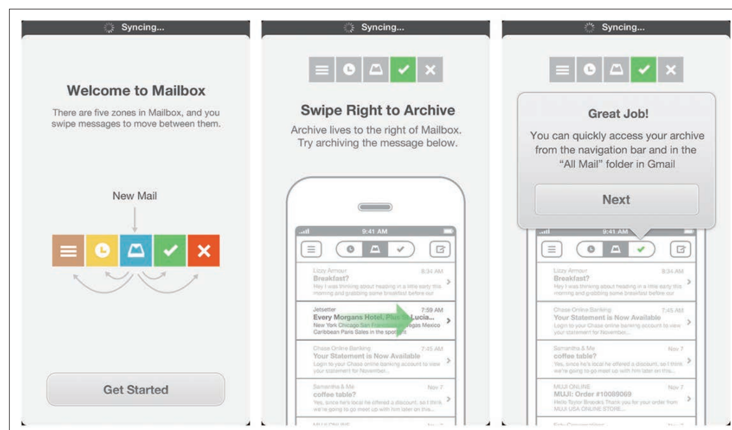



图 7-6
iOS 版 Mailbox：引导页的文字说明鼓励用户边操作边学习（http://www.youtube.com/watch?v=URd0j5vEsHE）

## DigiCal和Fantastical

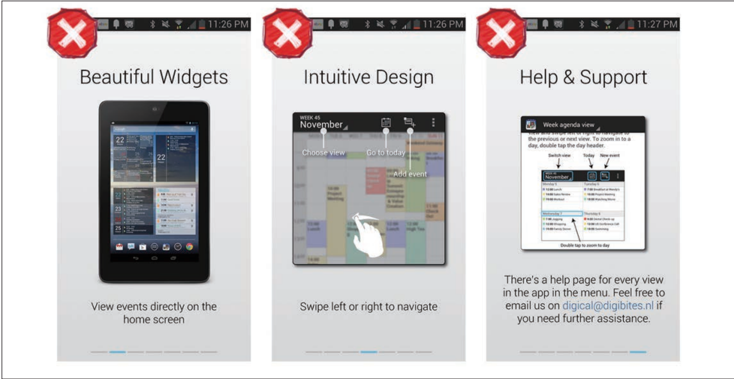



图 7-7
安卓版 DigiCal：文字与其描述的操作分离

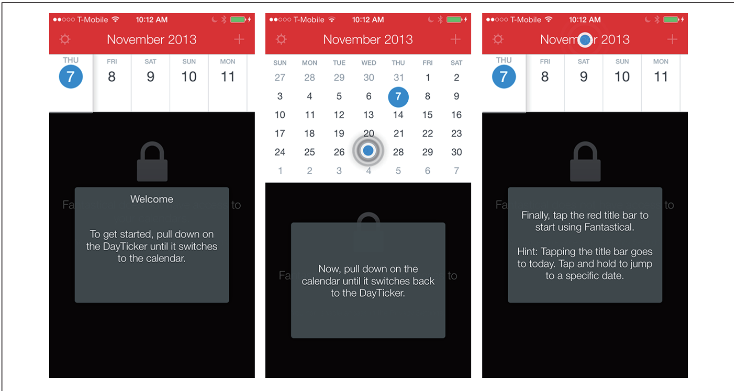



图 7-8
iOS 版 Fantastical：文字说明引导用户用手势操作（http://www.youtube.com/watch?v=e6Q8FbuNwiQ）

## Catch和Clear

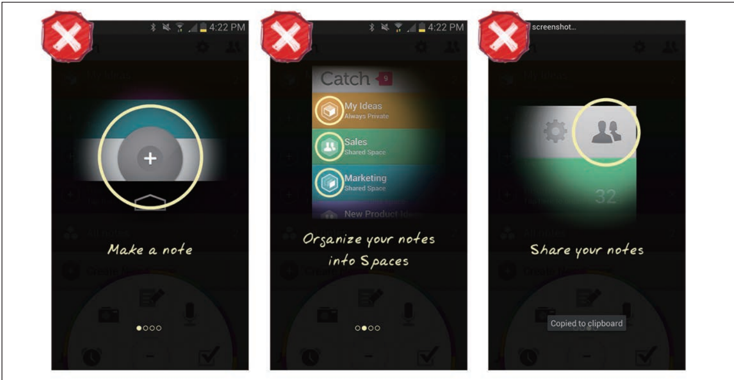
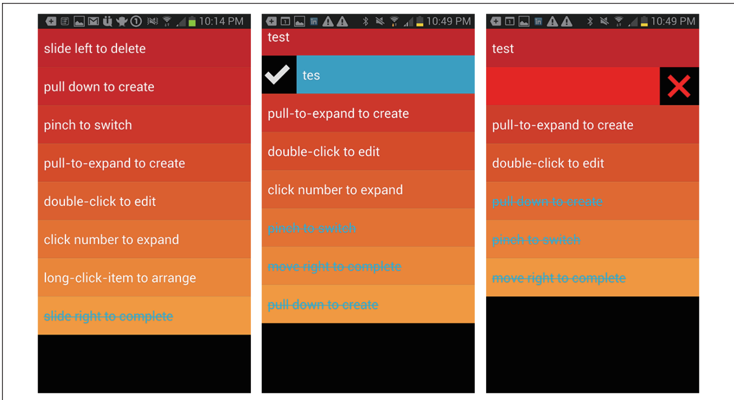



图 7-9
安卓版 Catch：全局游览描述了功能和操作，但没有让用户试用

图 7-10

安卓版 Clear：默认视图预先设置了任务列表，让用户边做边学习如何使用（http://www.youtube.com/watch?v=v3gGTG003_A）




## SlideStory和Vine

图 7-11

iOS 版 SlideStory：这个引导有图示也有讲解，但没有让用户实际操作

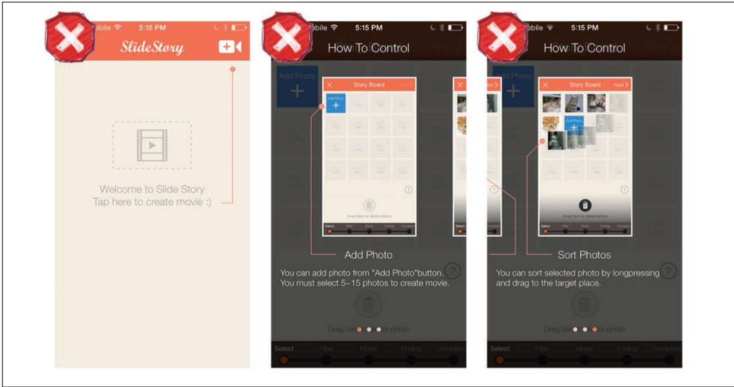
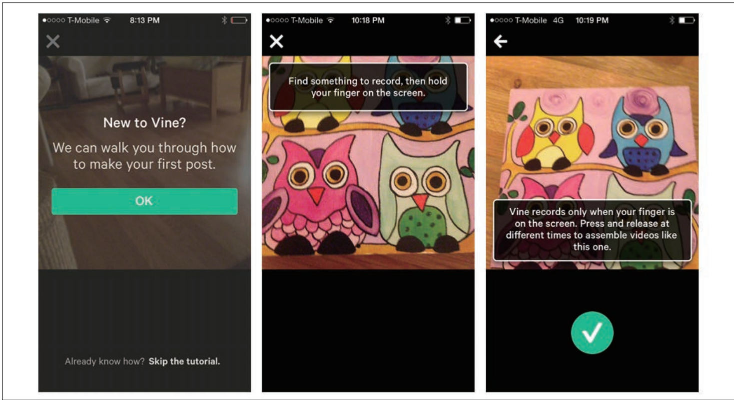



图 7-12

iOS 版 Vine：有图示和文字介绍，还鼓励用户亲自上手操作（http://www.youtube.com/watch?v=vBOC9DpC5ns）



## 原则2：不要前置引导

《额外加分》节目在《教程 101》中说道："如果你一上来就进行讲解，事无巨细地引导你的玩家，他们就会被信息的海洋淹没，没有动力玩下去了。"你只需把这句话里的"玩家"换成"用户"，其相关性对于应用设计师来说是显而易见的。

对于我们大多数人来说，这一点很难。毕竟，我们辛辛苦苦地设计出有用功能，自然希望用户能够马上知道这个应用都能做什么。

抑制住这样的冲动。正如《教程 101》视频中指出的那样，前置引导会熄灭用户的热情，让他们跳过你的引导。而且，用户一旦跳过引导，就不大可能从你的应用中有所收获，也不大可能向其他人推荐你的应用。

在用户需要时为他们提供简单易懂的信息，而不要让他们被那些等到用时几近忘光的信息所淹没。记住，你在营造第一印象。为什么不在初见之时留给用户意犹未尽的感觉，而要让他们就此止步呢？

限制前置引导，只对主要功能做个初步介绍，以便让你的用户潜心进入，快速上手。后面碰到的视觉吸引能激发他们探索更高级的特性和功能。

让我们对比几个掌握了要领的应用与不得要领的应用。

首先，我把 iOS 版 Phoster 和 Windows Phone 版 Creative Studio 放在一起进行对比。Phoster 采用了一张混乱的引导页面，上面塞满了用户需要了解的所有说明。无论何时，用户想要查阅一项操作，都必须费力地把通篇的说明阅读一遍。

对比来看，使用 Creative Studio 处理一张图片时，只有在用户选中了一个工具之后，才会显示出操作说明。在示例中，我打开一张图片，选择"聚焦"效果，然后我会被引导到一个两步处理流程。预览了处理效果之后，我决定继续编辑一下，在点击"编辑"项后，我会看到另一个相关的情景化提示。

[小贴士]

不要预先告知用户可能用到的所有信息，否则他们会不堪重负；把信息分成小块，当用户需要时再显示。

## Phoster与Creative Studio

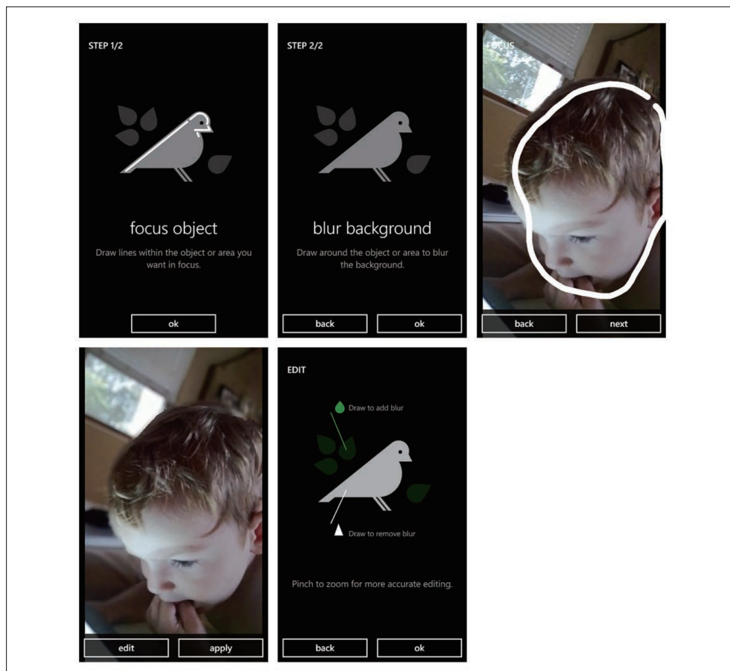
图 7-13

iOS 版 Phoster：尽管只有一页引导，但混乱的布局预先展示了太多信息




图 7-14

Windows Phone 版 Creative Studio 适时为用户提供情景化帮助



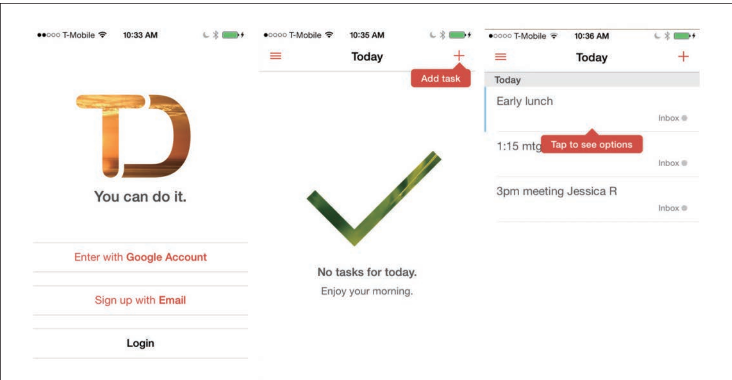
## Dooo与Todoist




图 7-15
iOS 版 Dooo：排山倒海似的前置信息没完没了（总共有 11 页）

图 7-16

iOS 版 Todoist：一个气泡提示邀请用户添加第一个任务，随后另一个提示介绍了菜单选项




## Buy Me a Pie!与OneNote

图 7-17

安卓版 Buy Me a Pie!：令人困惑的、脱离了情景的前置引导

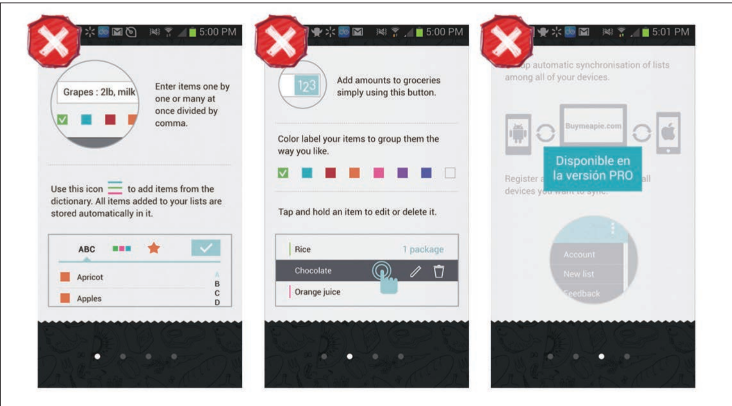
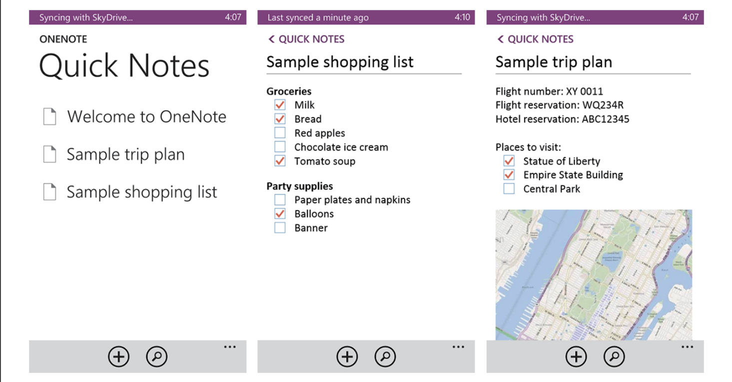



图 7-18

Windows Phone 版 OneNote：列出两个示例列表，使用户可以在实践中学习



## Clipchat与Kik

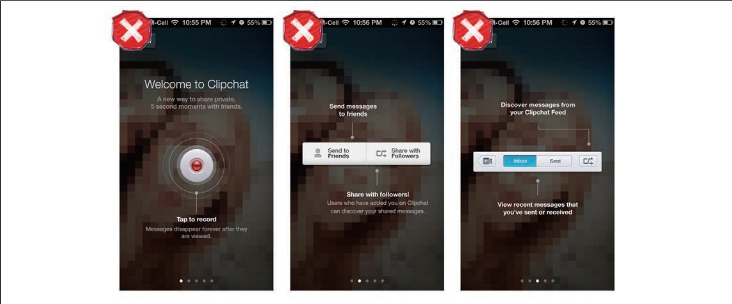



图 7-19

iOS 版 Clipchat：太多前置引导信息，脱离了使用情景

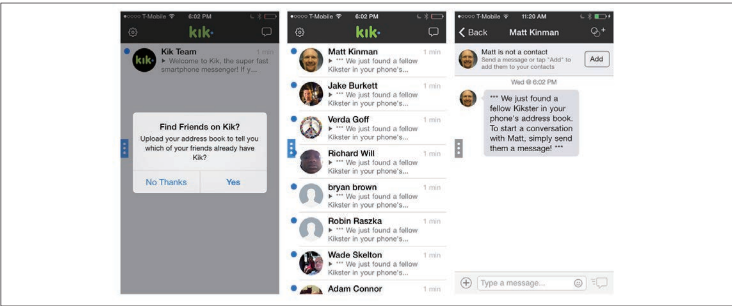



图 7-20

iOS 版 Kik Messenger：导入联系人后即可看到一个怎样与他们开始聊天的提示

# 原则3：给予用户成就感

实际上，你可能记得在《教程 101》中这个原则叫作“让引导更好玩”。但说真的，“好玩”并不适用于所有的应用。视频解说员说“你的游戏引导应该像游戏中的其他部分一样引人入胜”，我们可以触类旁通，意识到这条原则也适用于移动设计领域。

即使我们不能让新手的学习过程变得有趣，也仍然有一些方法让这一过程富有成就感，并使之完全融入到应用的整体体验当中。一个比较好的做法是实时交互，让用户实际完成操作。这样做能产生一种巩固了所学内容的感觉。

就算“好玩”不是我们的目标，但有时在引导中融入一种趣味性也很不错。下面，我们首先对比 NBC News 与 Flipboard。这两个应用对此采用了不同的方法，后者是正确方法的典型范例。

## NBC News与Flipboard

尽管 NBC 的五个幻灯片引导页的字体都挺“有趣”，但仍然让人觉得

像本教材。相比之下，第一次打开 Flipboard 就有种截然不同的感觉。

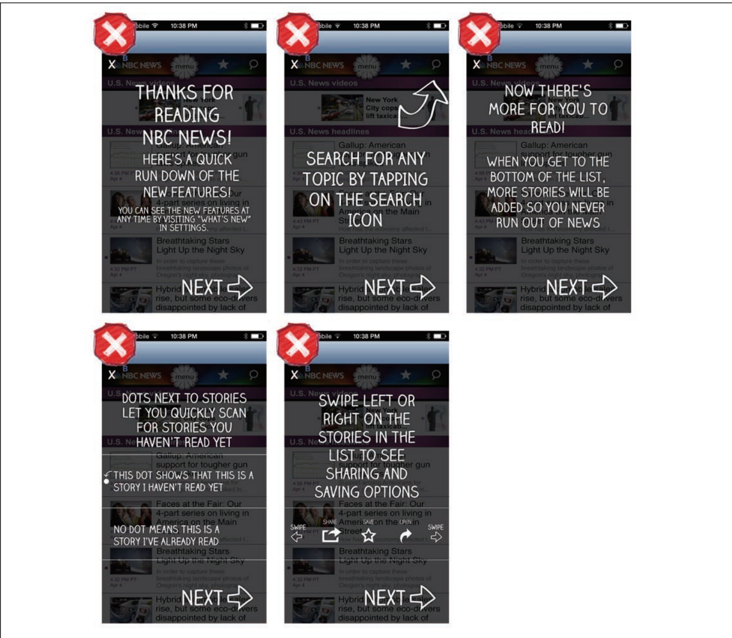



图 7-21

iOS 版 NBC News：采用了一种有趣的字体，但文字冗长得让人提不起兴趣

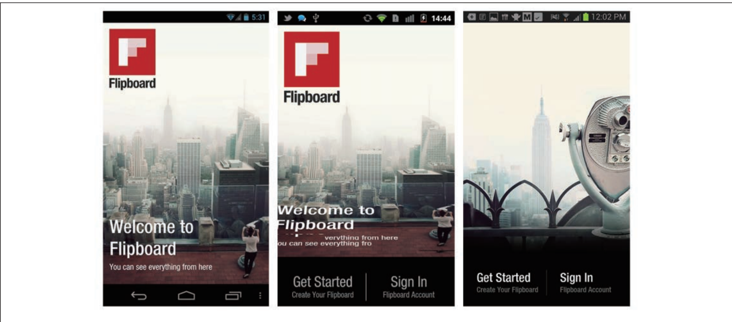



图 7-22

安卓版 Flipboard：用有趣的嵌入式提示吸引用户，并强化了这种用于应用导航的关键手势（http://www.youtube.com/watch?v=rQASZtuvBCs）

Flipboard 没有文字说明，但页面下半部分顽皮地向上翻起，挑逗你在翻到下一页前瞄一眼那里的内容。再看一次，你可能就明白了那个提示的含义—向上滑动查看隐藏的内容。但如果此时你没有滑动，就会出现一个嵌入式的视觉吸引，写着“向上滑动以继续”。

随后的每个翻页转场都强化了新用户在探索 Flipboard 内容时需要学会的滑动手势。这既有趣，又有成就感。

## Noom和DailyBurn Tracker

安卓版 Noom 是一个不可思议的应用，在使用给予成就感机制深度吸引用户方面做得非常不错。体验设计从头至尾都很诱人，使得用户会反复使用应用。Noom 采用了“升级”的游戏化技巧（参阅第 8 章），激励用户成功完成一系列任务，达到下一级所需水平。

它还运用了一个让应用始终有趣的“惊喜”元素。完成程序设置后，我马上获得了一个“优胜者奖”，激励我再接再厉。

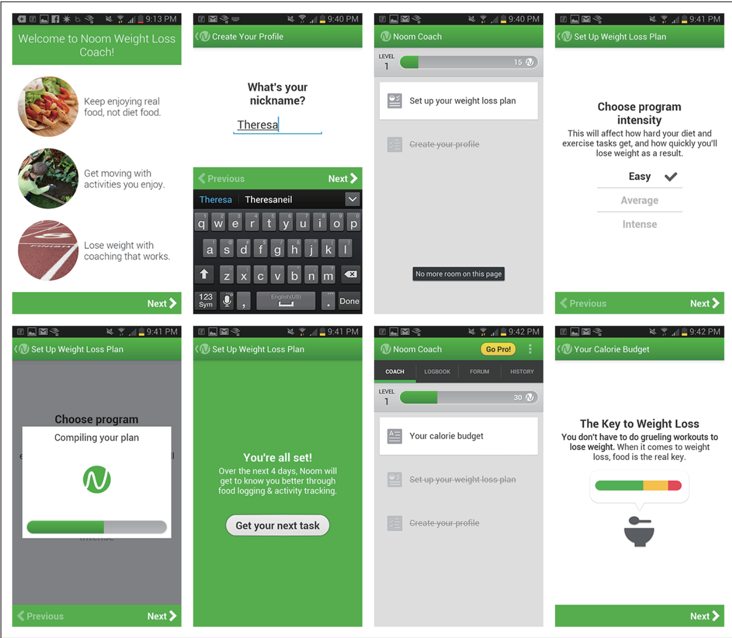



图 7-23

安卓版 Noom：引导设置轻松友好，始终伴随着正面反馈

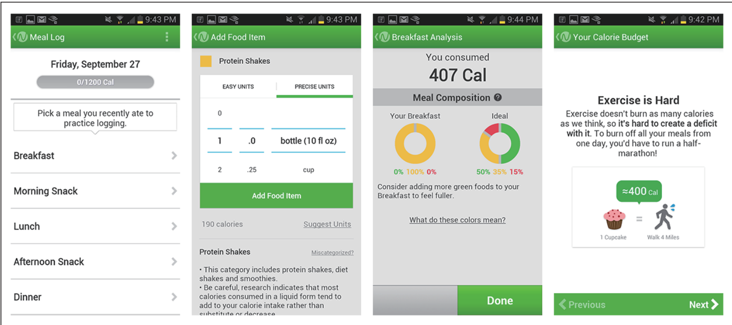
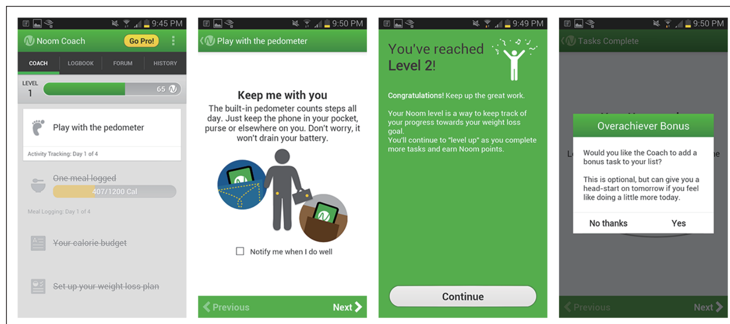



图 7-24

安卓版 Noom：为确保用户理解，将任务设计成参与式（比如练习记录一餐所吃的食物）

图 7-25

安卓版 Noom：任务开始时容易完成，但难度随着进度加大；升级让体验游戏化，也带来更大的成就感




**[小贴士]**

如果你现在下载安卓版 Noom 的话，可能会发现最新版与上面的截图（2013 年 10 月）有些不同。我很好奇是什么促成了这些变化，于是联系了 Noom 的项目副总裁 Eli Holder。

Eli 说，用户测试显示，2013 年版的设计让参与者感到“精疲力尽”。改版的目的是把着陆流程分割成若干小步骤，使之不大像一个任务列表，而是更贴近用户的真实减肥经历。

今年请继续关注 Noom，因为 Eli 和他的团队改进了着陆流程，目的是更迅速地带给用户愉悦的体验，让用户与 Noom 之间建立起情感联系。

相比之下，设置 DailyBurn Tracker 的过程非常繁琐。一上来就有 6 个设置页面，然后让我直接进入有 10 个选项列表的首页。没有引导，没有激励机制，没有惊喜。（对于激励型应用，惊喜是很有用的。）

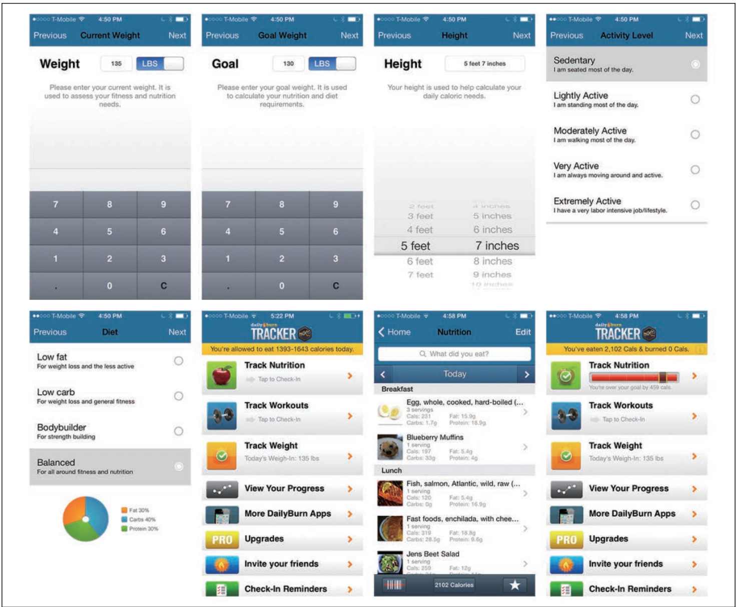



图 7-26
iOS 版 DailyBurn Tracker：设置过程就像做枯燥乏味的家务

[小贴士]

用你营造的成就感使应用更具魅力。可以考虑适时增加趣味性元素，并让用户通过动手练习获得“经营”感。

## 原则4：在使用中强化学习

还记得在科学课上，当通过一个简单的实验理解了一个概念的时候，发出“啊哈！”的感叹吗？这就是我们现在所要说的情况。当然，老师已经讲解过那个概念，但是，你正是通过自己做实验才真正学会了它。

同样的观点也适用于引导。如果你遵循了前三条原则，那么这个强化过程就会是自然而然的了。很简单，比如在引导中讲解一项操作时，附加一个微小的视觉或听觉上的反馈。然后，当用户以后进行这项操作时，同样的反馈会强化他学到的东西。（如果你确定使用声音反馈，记着提供一个关掉声音的方法，以免用户在熟悉了你的应用之后对音效产生反感。）

这条原则是对“不要前置引导”原则的补充。不要试图一次性把你的应用一展无余，可以考虑设计一种引导用户循序渐进的浸入式应用体验。还可以随着时间的推移展示更高一级的功能，或者发放意外“奖品”以资鼓励。通过这些方式，你会帮助用户在使用应用的同时强化学到的内容。

[小贴士]

学习不是一次性的；在使用应用的过程中，应该巩固用户在引导中了解到的知识。

图 7-27
iOS 版 Polar：做了几个问卷调查之后，我看到了一条提示；iOS 版 DuoLingo：参与即奖励宝石

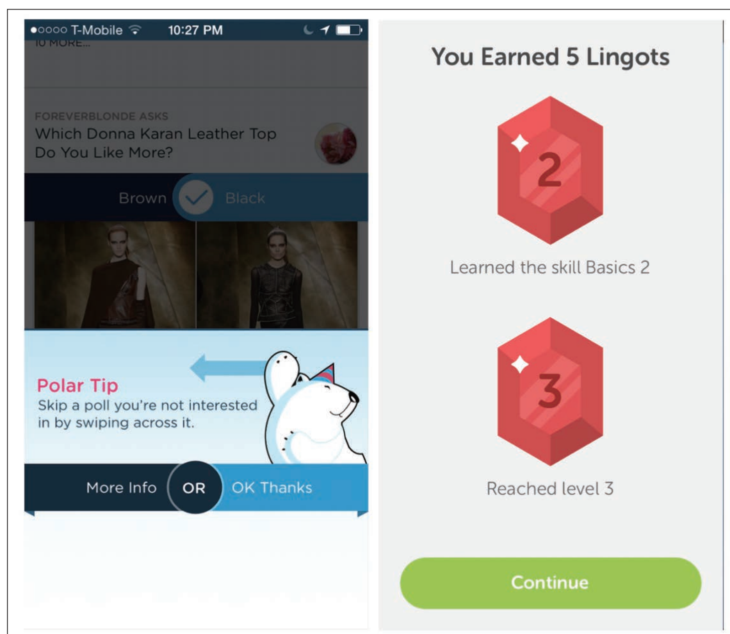
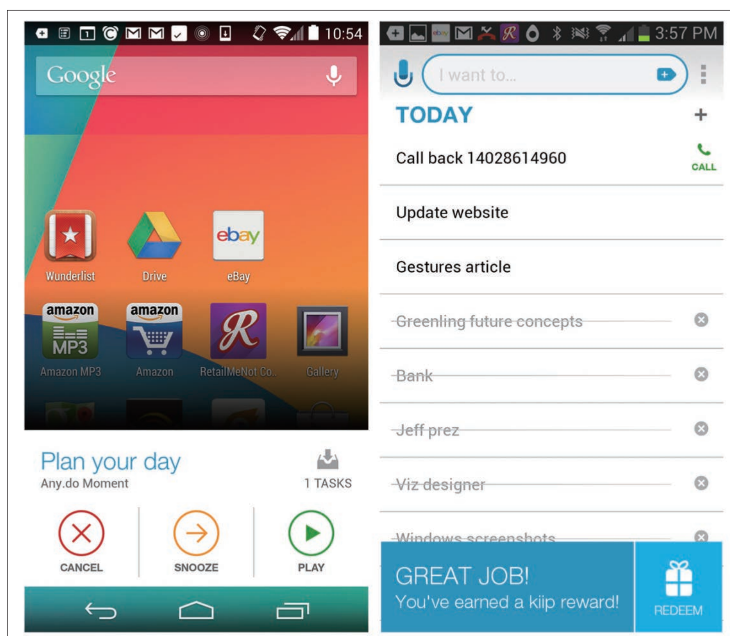



图 7-28
安卓版 Any.do：用小部件鼓励用户进行互动，完成任务时自动给予奖励



## 原则5：倾听用户的心声

想一想，你潜心开发并打磨你的应用已达数月之久，那么谁是向用户阐述其操作程序的最佳人选呢？

可能不是你，根据《教程 101》中讲解员的说法："你身为一名设计师，已经为一个项目工作了一两年之久，这时候你很容易把其实非常费解的难题看成直观或显而易见的事情。"

长期接触一个应用会使我们弄不清用户使用时需要先掌握哪些要领。关于这一点，我们会在第 10 章中进行讨论。但是现在要记住，在衡量你的引导效果的时候，只有你的用户（或游戏领域里的玩家）才是唯一可靠的人选。

[小贴士]
对自己的应用了如指掌，有时反倒不知道如何指导新手用户。做个不带倾向性的用户测试，问题就能迎刃而解。

简单适当的用户测试能暴露出应用使用过程中的任何障碍。观察你的用户，看看他们在哪里绊住了，在哪里碰到了问题。倾听他们与应用互动时的评论，稍后再向他们提问。如果在他们使用应用的时候提问，你很可能会不自觉地引导他们说出你想要的答案。

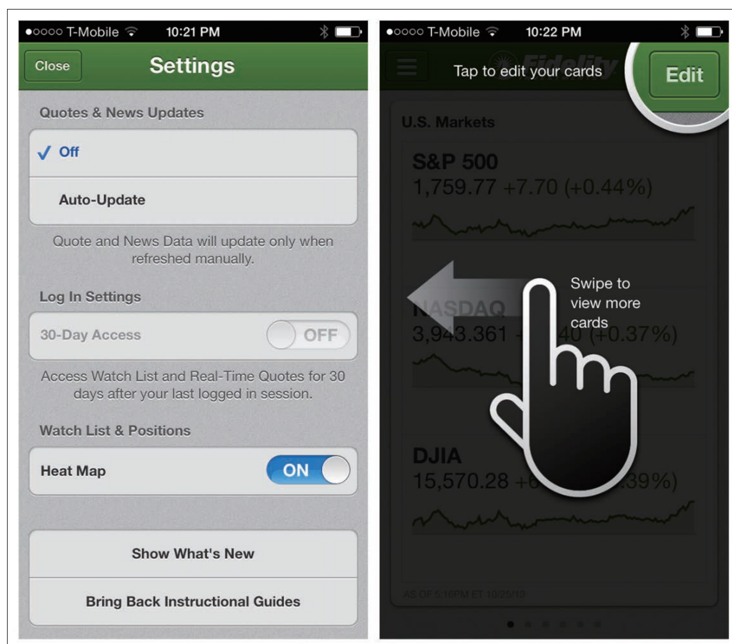
这就引出了《额外加分》专家列出的后几个技巧：

- *不要到最后再设计引导。*
  你必须从一开始就把引导融入整个设计过程。很多团队把引导放到最后再添加，这实在是大错特错，甚至有可能损害你的应用的命运。本章指出的许多反模式引导页可能就是由于事后设计造成的，这些应用本身设计得并不差。引导应该被当作应用设计中最重要的一个环节。如果引导失败了，你的应用也就失败了。

- *引导应该允许用户跳过。*
  你有没有用过每次打开都强迫你看完整个引导的应用？如果有，那么这个应用你可能用不长。

- *引导中传达的所有信息应该都可以随时访问。*
  同样，引导中提供的任何帮助信息都应该作为参考，令用户随手可得。比如说，Fidelity 让你可以在"设置"里重新启动引导。参见第 10 章查看更多示例。

图 7-29
iOS 版 Fidelity：你可以点击“Bring Back Instructional Guides”，重新打开引导




## 视觉吸引模式

不管你的应用是否一定要有着陆引导，你都有可能想在应用中吸引用户使用某个功能。视觉吸引还有助于在某个特定的流程中向用户推介新的功能，而不是在启动时仅仅显示一个带有新功能列表的对话框。

### 提示

提示应该指向屏幕上某一个特定的工具或操作按钮。要避免使用幻灯片提示（Zappos）或对话框（HoursTracker），myAppFree 那样一屏多个提示的情况也应该避免。至于提示的巧妙设计，可以参看 Evernote Food 和 Mint 的示例。

［小贴士］

把提示放在操作按钮或菜单项旁边，内容要简短，并在互动开始时（例如点击屏幕）隐藏提示。

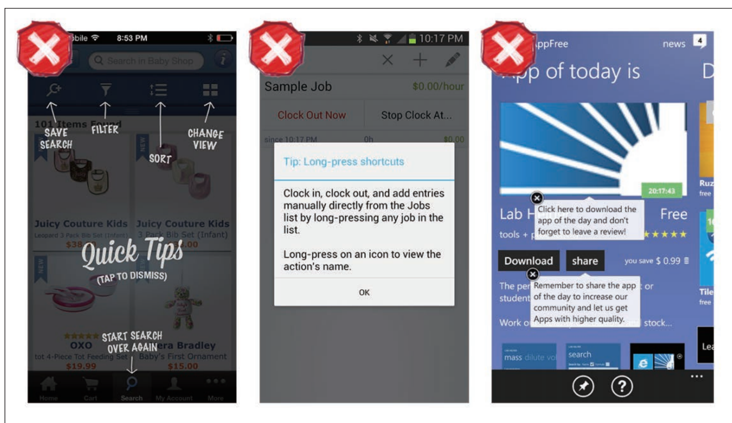



图 7-30
iOS 版 Zappos、安卓版 Hours-Tracker 和 Windows Phone 版 myAppFree：差劲的提示设计

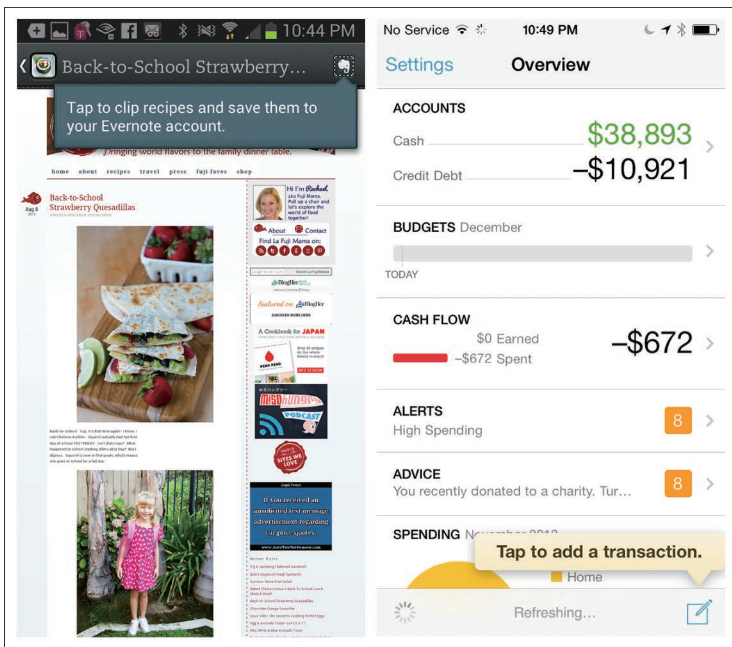



图 7-31
安卓版 Evernote Food 和 iOS 版 Mint：优秀的提示设计

要想了解如何设计高效的提示，请参阅本章“番外篇”中关于 Intuit 的案例研究。

## 持续的视觉吸引

持续的视觉吸引模式在某些例子中看上去也许和提示差不多，但它们在持续性上还是不尽相同。这些视觉吸引要么始终可见，要么保持可见状态直到用户执行指定操作为止。

在 RetailMeNot、Expense Manager 和 Pnadora 的例子中，视觉吸引一直持续到用户执行指定操作（分别是保存优惠券、记录花销和添加喜欢的艺术家）。点击操作后，视觉吸引就永久消失了。

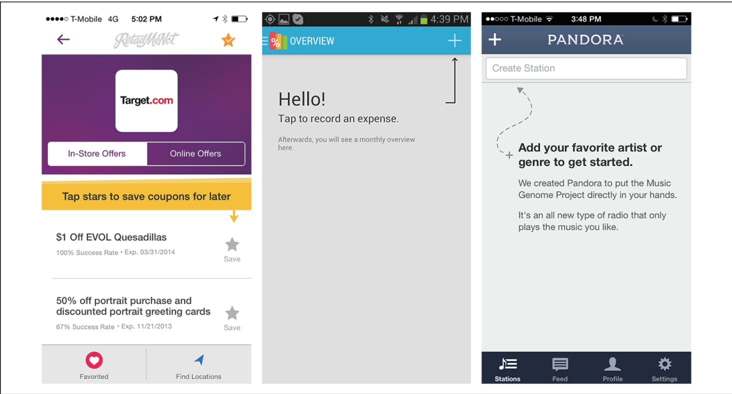



图 7-32

iOS 版 RetailMeNot、安卓版 Expense Manager 和 iOS 版 Pandora：持续的视觉吸引

但是 Pinterest 的视觉吸引是一直可扩展的。不管你添加了多少画板，总有一个视觉吸引，让你可以再添加一个画板。你要决定把视觉吸引放在哪个位置：在 Do It（Tomorrow）应用中，视觉吸引位于列表上方；在 Flipboard 中，视觉吸引则位于最后。

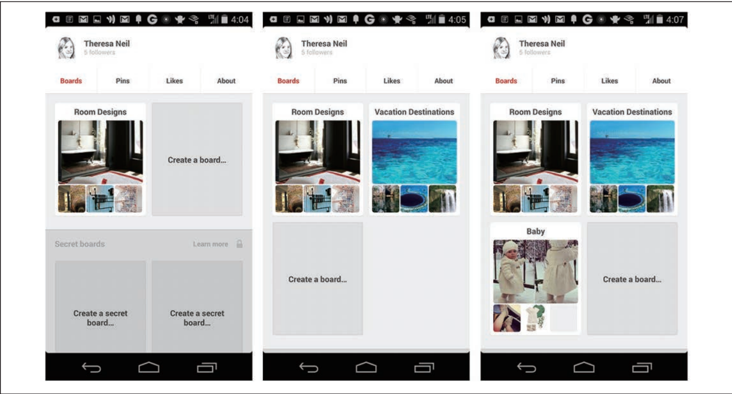



图 7-33

安卓版 Pinterest：持续的视觉吸引

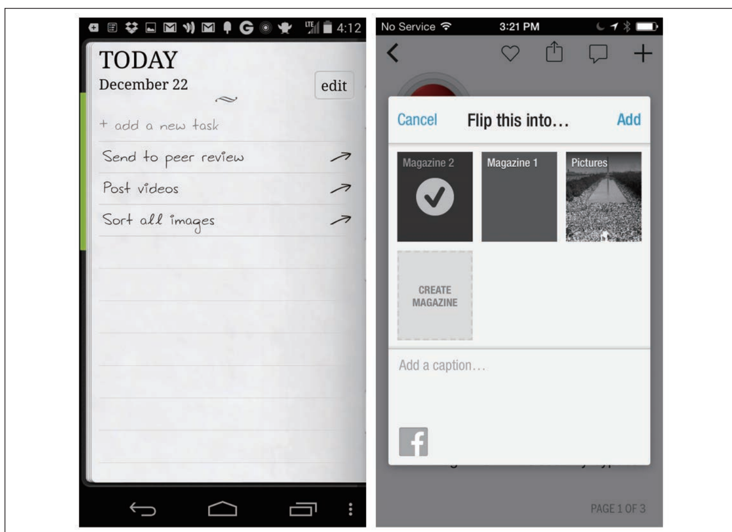



图 7-34

安卓版 Do It（Tomorrow）和 iOS 版 Flipboard：视觉吸引可以放在列表开始处或末尾处

Argus 有一套内置逻辑，可根据用户预先设置的锻炼与营养目标显示持续的视觉吸引。比如，“点击设置体重”（Tap To Set Weight）的六角形视觉吸引按钮会一直显示在屏幕上，在我记录了今天的体重之后，就会有一个显示出我的体重的六角形取而代之。每周我都会看到同样的视觉吸引。

[小贴士]

想一想如何设置视觉吸引的恰当时长。是应该始终可见，还是在操作后隐藏，又或是在特定的情景下显示？做个用户测试，确保视觉吸引对他们确有帮助，而不是在打扰他们。




图 7-35

iOS 版 Argus：设置体重的视觉吸引根据用户设定的目标重复出现

## 可发现的视觉吸引

可发现的视觉吸引似乎有点自相矛盾，但它是在不扰乱屏幕的前提下，能鼓励用户进行特定交互操作的一种行之有效的方法。这种视觉吸引故意设计成在用户执行了常见手势操作（如点击、下拉或滑动）之后才显示出来。

Polar 与许多数据流应用类似，下拉内容列表时会看到“松开即可刷新”的文字视觉吸引。点击 SigFig 图表（想查看大图时的一个自然手势），会提示用户旋转设备至横屏。

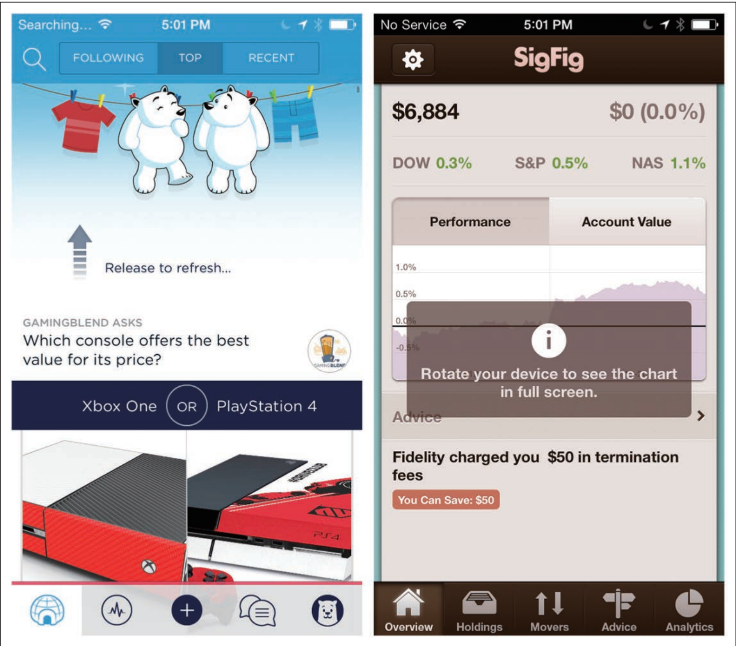



图 7-36
iOS 版 Polar 和 SigFig：可发现的视觉吸引

## 番外篇：视觉吸引——铺上欢迎毯

作者：Alissa Briggs，Intuit 公司用户体验经理兼主任设计师



想象一下，一位新结识的朋友邀请你去参加一场晚宴。你提着红酒到了晚宴门口，但在进门之前你犹豫不决。里面有你认识的人吗？要是你出了个糗，让自己丢了面子怎么办呢？

就在这时，宴会的主人为你打开门，给你一个大大的拥抱，欢迎你赶快加入进去。他帮你拿着外衣，将你介绍给其他宾客。你的迟疑一扫而光，你微笑着，充满自信，准备好享受这个夜晚。

设计一个令新手满意的体验与做一个很棒的主人没有什么差别。目标都是一样的：帮助你的客人快速克服焦虑，让他们感受到热情并充满自信。

刚开始设计一款智能手机薪资计算器 Intuit Snap Payroll 的时候，我们首要的任务就是打造一个友好的新手体验。我们的客户是一些刚刚接触工资报表的小企业业主。在精确计算薪资问题上，他们有太多的顾虑和担忧。为了表明这款计算器定能让他们掌握这项高难度的工作，我们必须要设计出热情友好且能激发自信的新手体验。

下面来看看，我们是如何找到解决方案的，以及我们在此过程中学到了哪些东西。

## 迭代欢迎体验

为了了解哪一种设计方案对激发信心最有效，我们收集了用户对各种设计理念的反馈。下面的图片是我们测试时用到的真实框图。

### 设想1：直接沉浸

我们的第一个设想就是把“欢迎毯”统统丢掉，让用户直接进入到他们今后会不断用到的应用功能菜单。

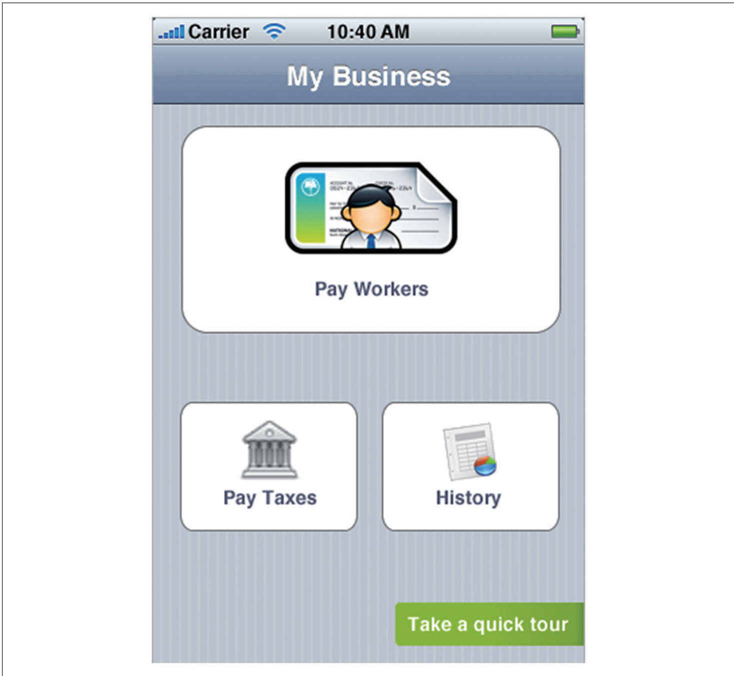



图 7-37
没有表示欢迎：让用户直接进入主页面

客户对此感到茫然无措。他们不确定应该先选择哪一项，并对独立完成工资计算感到更加焦虑。让我们继续晚宴的比喻，这种情况就像是我们的客人神奇地出现在聚会会场中央，没有熟人，主人也不知所踪。我们还需要提供一个迎接他们到来的场景。

### 设想2：使用向导

为了增加一个场景，我们使用了一种常见的视觉吸引模式—使用向导。我们测试了下面展示的 4 个版本，想看看究竟哪种模式最有效。




图 7-38
优点陈述：一张图片，简短的优点陈述和鲜明的行为召唤操作按钮

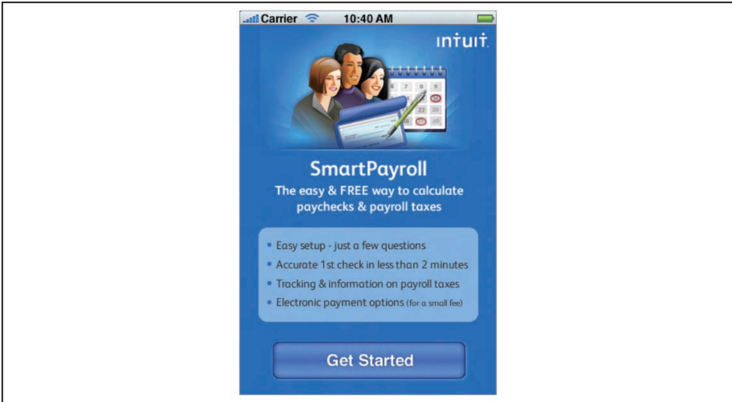



图 7-39
功能列表：优点陈述和简短的核心功能描述

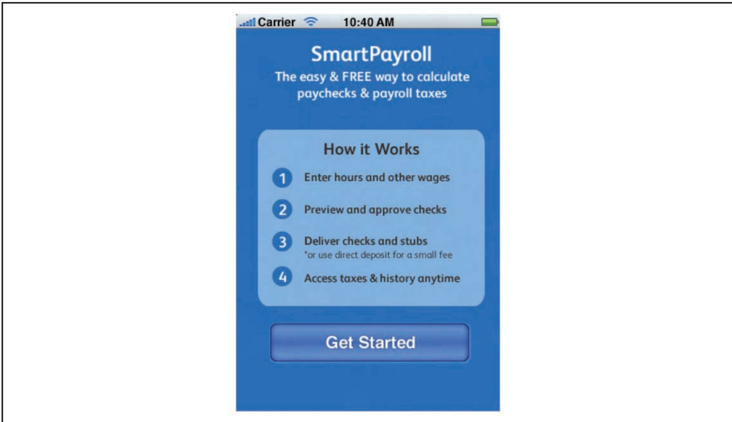



图 7-40
步骤列表：一步一步地说明如何使用应用

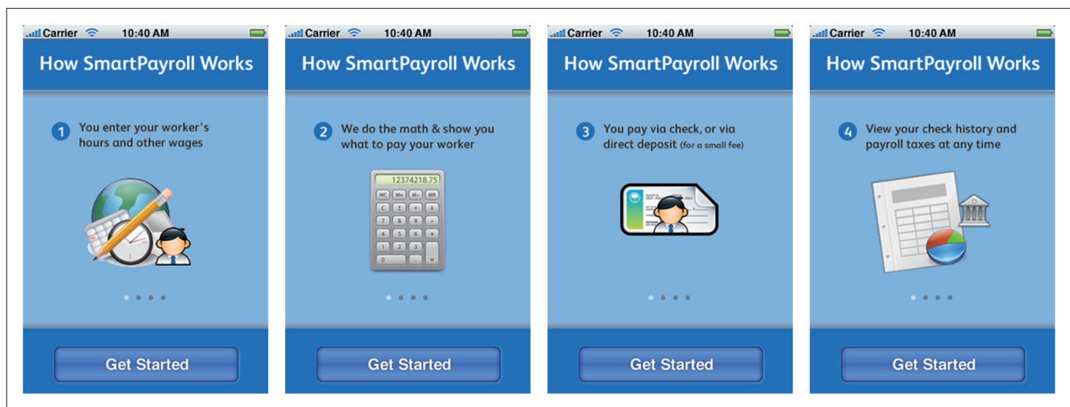



图 7-41　多步骤使用向导：一组介绍页，用户可滑动页面了解更多的应用信息

用户对这些方案反响一致：没有人阅读页面文字或是滑动查看功能列表。反之，他们快速跳过这些引导，直接点击“开始”。用户解释说这些页面挺碍事的，他们更愿意立刻直接体验应用本身。

提供前置引导传达信息的办法并没能帮助用户树立信心。这就像晚宴宾客到了我们的门口，却只看到一张便条，上面写着晚宴将如何进行并邀请他们自行进入。我们需要设定恰当的引导，不能碍手碍脚。

### 设想3：多步骤幻灯片

为了提供不打断用户操作流程的引导，我们尝试了另一种常见的模式—幻灯片。运用这种模式，用户可以看到真实的应用界面，但界面上覆盖了一层文字说明。

我们尝试了几种浮层方案，其中的两种如下所示。

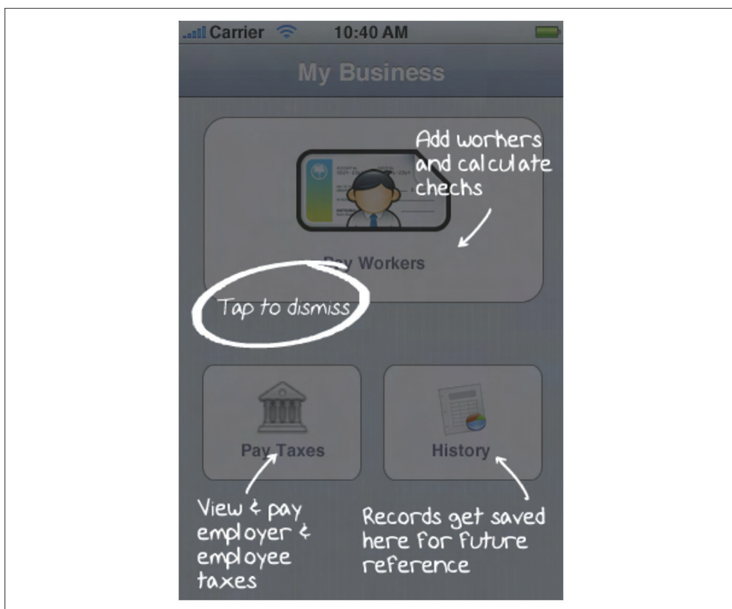



图 7-42
多步骤幻灯片：在主导航页面上突出多个功能点

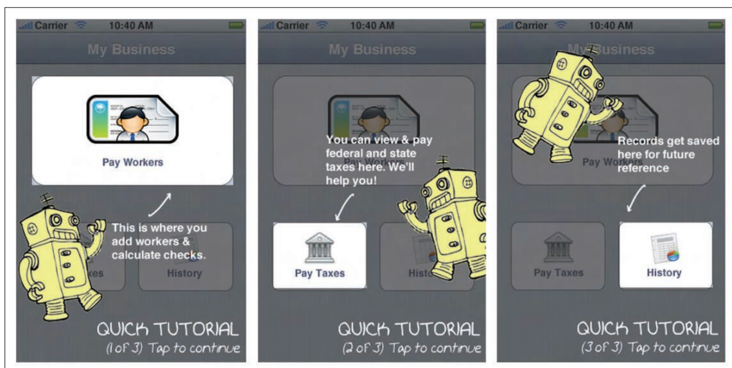



图 7-43
多步骤幻灯片：用户需要点击一系列步骤，每个步骤描述一个导航元素

虽然用户喜欢在背景上看到应用，但是对页面上那么多说明感到恼火。有些人评论说“我没有时间去费劲弄懂它”，“我就是想让这些东西别挡着我”，当然还有“我非得看完这些东西吗”。

我们离好的设计越来越近了，但问题是一次性显示太多说明会让客户觉得要学的东西太多了。这就像我们的晚宴主人迎接客人进来了，又向他引荐了 10 位他非见不可的客人，然后任由他们自己晕头转向地站在大厅里。我们怎样才能让体验更简单呢？

### 设想4：单步骤幻灯片

我们采用了幻灯片设想，但只关注用户首次进入应用时应该做的唯一一件事。

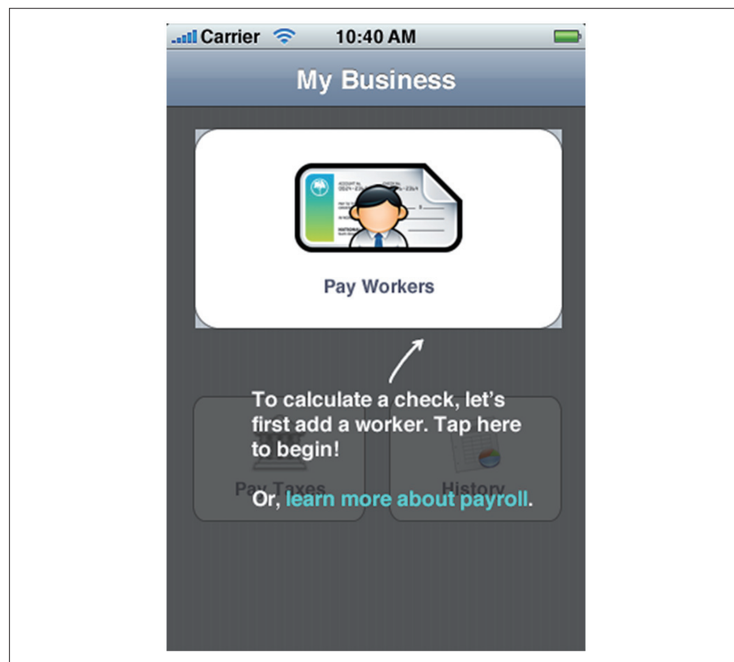



图 7-44
单步骤幻灯片：只说明最重要的一件事情

客户对这种设计喜爱有加。他们希望能察觉到应用的全部功能列表，但也希望得到从哪里下手的明确建议。

但这个模式还是让用户有点紧张。客户问道："既然你们有一个引导，这是不是就意味着这个应用对我来说太复杂，我不可能自己搞明白？"我们也观察到，用户试图点击灰色背景关闭浮层，想看清楚浮层下面是什么。我们看到了进一步简化设计、避免干扰用户的机会。

### 设想5：提示

在这个设想中，我们保留了单步骤幻灯片的有效部分（一个清晰的行为召唤按钮和一个可见菜单），同时修正了一些问题（引导链接和灰色浮层）。用户在这个设计里看到的是一个提示样式的行为召唤操作，并且可以点击屏幕任一位置隐藏它。




图 7-45
提示：一个行为召唤操作，屏幕其他部分均可点

这种模式的测试结果非常理想。客户不仅明确地知道第一步怎么做，也知道自己处于哪个位置，以及哪些功能对他们可用。我们终于可以让我们的晚宴宾客们感受到热情和自信，与此同时告诉他们："菜谱上有一道菜你一定要知道！"

在为用户打造一个完美的新手体验的过程中，我们学会了把以下两个原则牢记在心：

- 让开道

  不要让用户界面上有含混不清的元素，不要强迫用户耐着性子看完使用向导，否则会让他们感觉你在隐瞒什么。

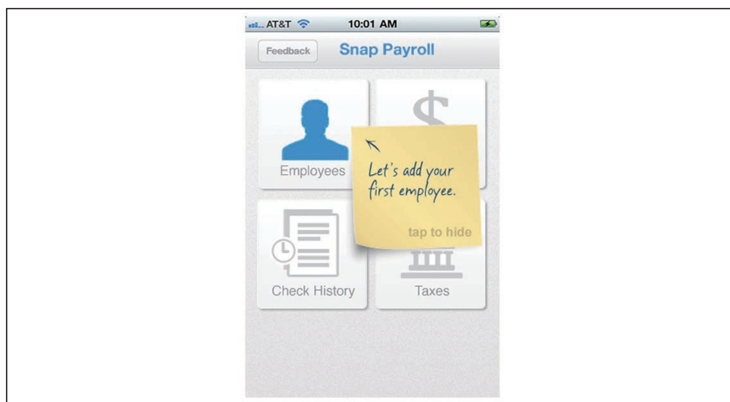
- 提供引导

  提出从何起步的建议，理清流程，消除不确定因素。

这些研究结果指导我们设计出了 Intuit Snap Payroll 新手体验的最终版本。正如设想 5 中所说，我们提供了一个提示，指向用户应该执行的第一步操作。

当用户第一次打开应用时，会看到主页上放着一个友好的便利贴，指向推荐的菜单项。但他也可以看到其他的菜单项，也可以选择点击隐藏便利贴。

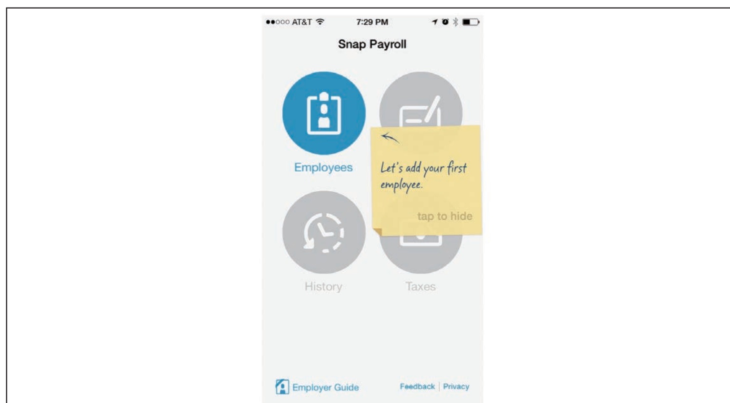
图 7-46
iOS 7 之前的设计版本：可隐藏的便利贴




这一设计非常成功地将客户的恐惧转化为自信。用户很快就理解了应用提供的各项功能，同时也感到自己知道接下来为获得积极的体验应该如何去做。

我们根据 iOS 7 设计指南，使用了一点幻灯片效果和动画效果，实现了开放性和引导性之间的平衡。在这个示例中，菜单首先加载进来，然后便利贴渐出，传达了有用的信息，也没有妨碍用户。

图 7-47
iOS 7 的设计实现：有用但不碍事




## 总结

在吸引客户进入你的应用时，不要仅仅指着大门，希望他们自己走进来。要花时间精雕细琢，设计出一个体贴周到的体验，让用户感到友好、自信。这样他们会期待下一次的造访。

第 8 章

# 社交模式




- 社交模式

  社交注册（Social Registration）、连接（Connecting）、关注（Following）、个人资料（Profiles）、群组（Groups）、游戏化（Gamification）

提供社交连接的服务仍在与日俱增，并且进一步细分。现在我们可以通过分享美食、购物、健身以及许许多多的事物，与有共同喜好的人相互连接。在这一章里，我们会详细分析一些社交模式，这些社交模式可以帮助你的用户通过移动设备建立这样的连接。

## 社交注册

在第 2 章中，我们讨论了注册表单模式。在此，我们将具体地讨论社交应用中的用户注册流程。

## MapMyFitness与We Heart It

MapMyFitness 提供了 Facebook 登录，用 Facebook 的 API 接口显示一个单独的授权页面。如果用户授权，它就会从 Facebook 的个人资料中把相关的注册信息调取出来，用以创建你的 MapMyFitness 个人资料。

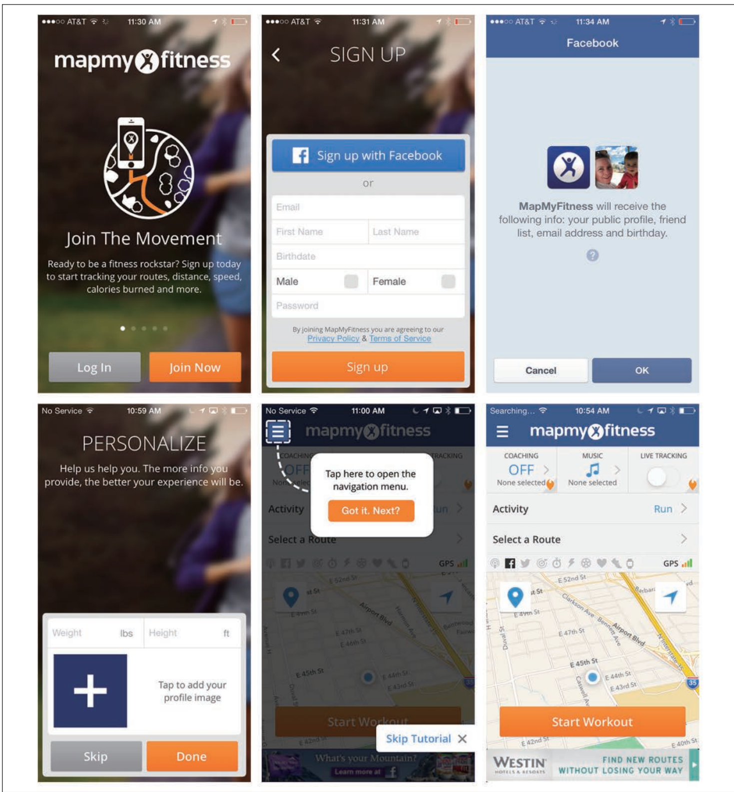



图 8-1
iOS 版 MapMyFitness 通过提供 Facebook 登录和选填个人信息简化了注册流程

遗憾的是，We Heart It 像其他许多应用一样，没有正确使用 Facebook API 接口实现快捷登录。在我用 Facebook 账号授权注册之后，We Heart It 要求我登录 Facebook（实际上我已经登录了），并且直到那时才告诉我，如果我想用 Facebook 注册，就必须修改我的 Safari 浏览器设置。到底是怎么回事？最严重的是还有一个 bug：“不发布信息到 Facebook”选项连续出现了好几次。就是这些混乱的多余步骤和磕绊让用户感到挫败，并最终放弃使用。

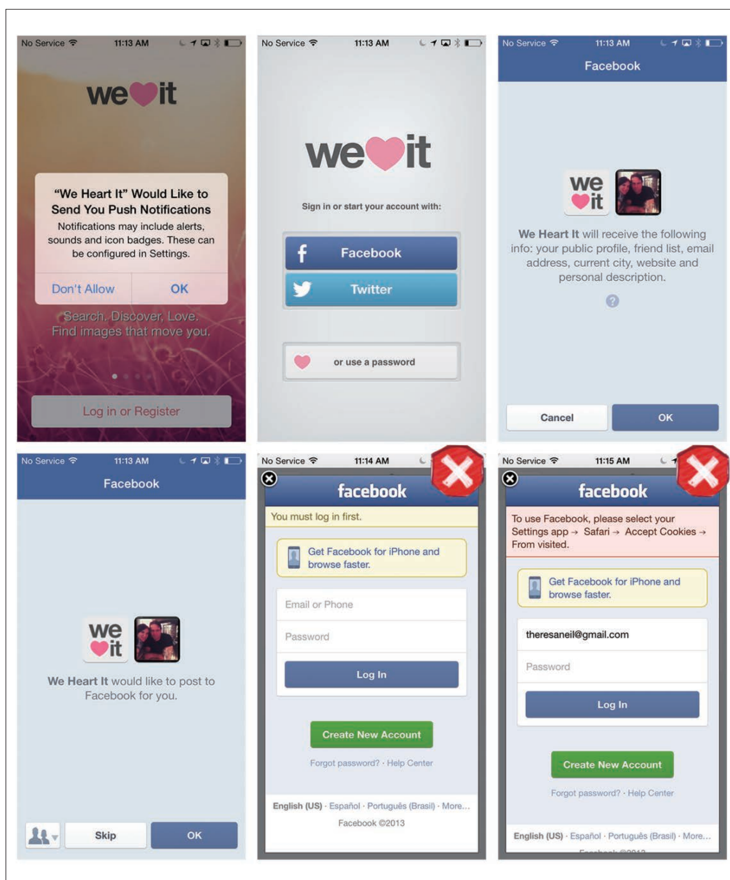



图 8-2

iOS 版 We Heart It：我不喜欢（heart）所有这些授权页面和错误

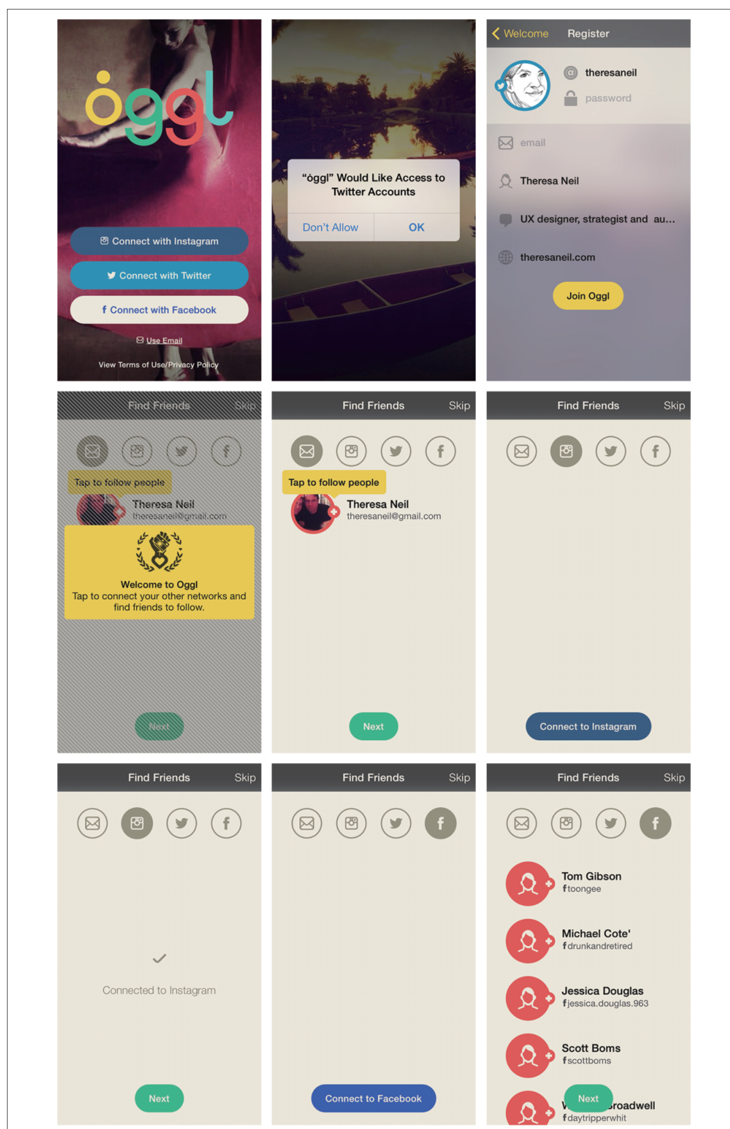
［小贴士］

正确使用社交网络的 API 接口以简化注册体验。一定要对注册流程的各个路径都进行质量保证测试。

## 连接

利用其他社交平台的 API 的一大好处是，它能让用户访问他们已有的社交连接。Oggl 提供了一种简单的连接流程作为注册过程的一个可选部分。在这个例子中，我用我的 Twitter 账号注册 Oggl，然后 Oggl 为我生成了一个预先填写了的个人资料页，我只需填写密码和邮箱地址即可。几次点击之后，我就能和我在 Instagram、Twitter 和 Facebook 上关注的所有人连接起来。

图 8-3
iOS 版 Oggl：加入其他社交网络的连接作为注册流程的可选项




LinkedIn 在网页版和移动应用上发布了一个新的流程，这个流程对推动用户彼此建立连接产生了极大的影响。任何时候你接受了一个邀请，都有一个全屏页面提示“查看在 LinkedIn 上你可能认识的人”。

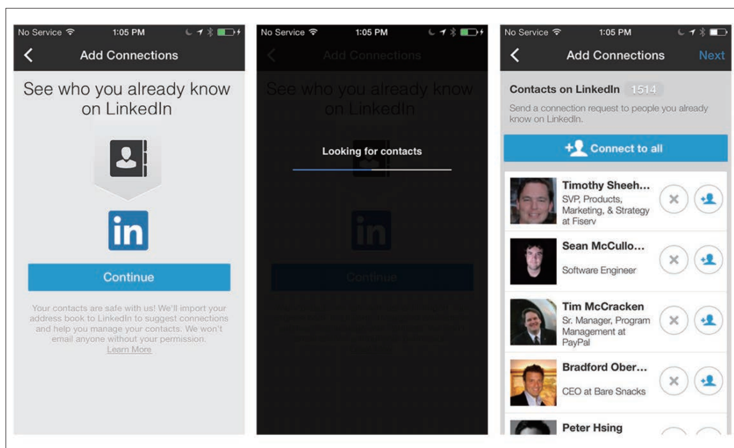



图 8-4
iOS 版 LinkedIn：接受一个邀请后获得与更多人连接的建议

让我们再来看看 MapMyFitness，我给它简单体贴的社交注册打了高分，但是它的连接功能没有放在注册流程当中。不过这倒不是什么问题，因为它在用户正常的操作流程中有计划地插入了连接提示：从主菜单到健身前后的视觉吸引，再到近期活动（Activity Feed）中的视觉吸引，贯穿始终。

［小贴士］

在应用中始终提供连接点。注册时在一个或多个其他用户操作流程中提供连接的机会是种不错的方法。

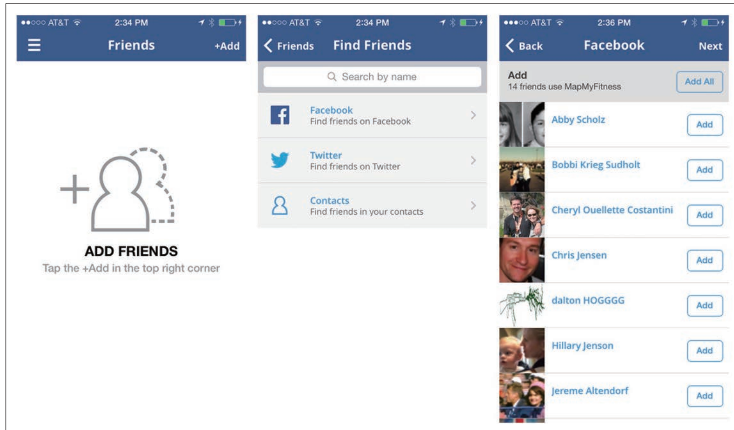
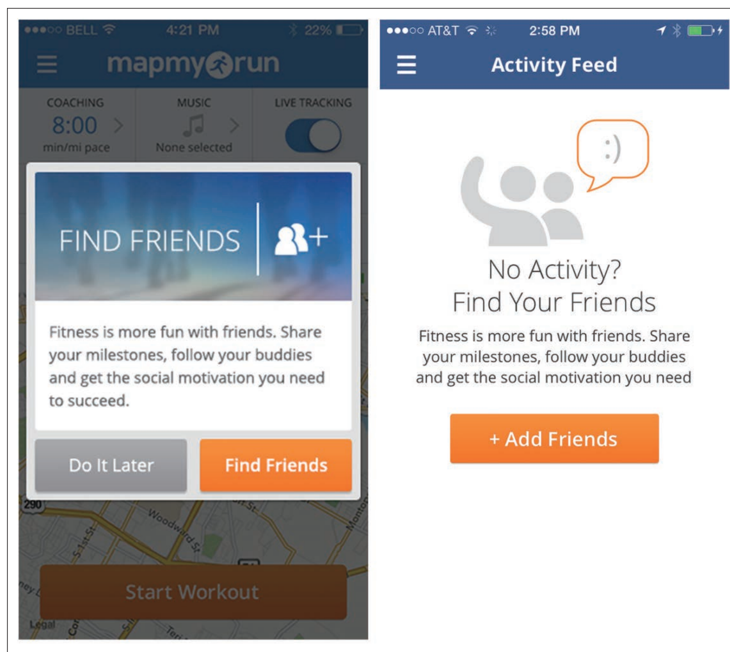



图 8-5
iOS 版 MapMyFitness：从主菜单中查找并添加好友

图 8-6

iOS 版 MapMyFitness：健身前后及近期活动中都有提示连接的视觉吸引




## 关注

前面所有的例子讲的都是如何建立批量连接。但你可能还想为用户提供一种一对一的连接选择。多亏了 Twitter，“关注”（Following）一词已经成了这种方式的通用术语。在 Twitter 和 Zoomingo 应用中，“关注”流程是这样实现的：我正在浏览，看到某个有趣的东西（比如一条推文或是一张优惠券），然后看看这条信息是谁分享的，再查看一下他还分享过什么。如果那些内容吸引我，我就会关注那个人。

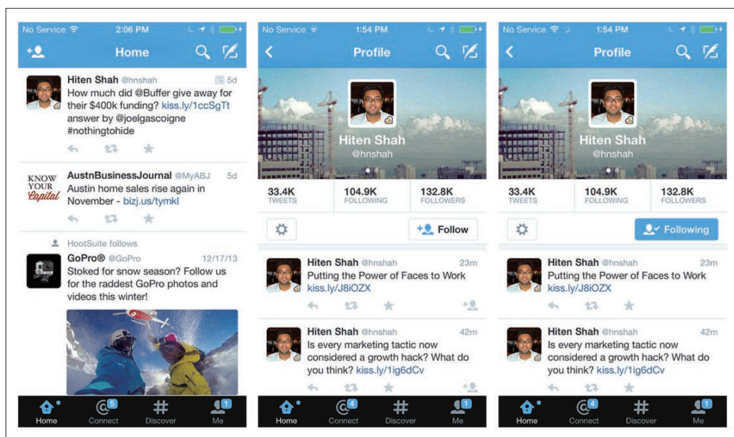



图 8-7

iOS 版 Twitter：“关注”流程可以让你看到某一用户分享的更多内容

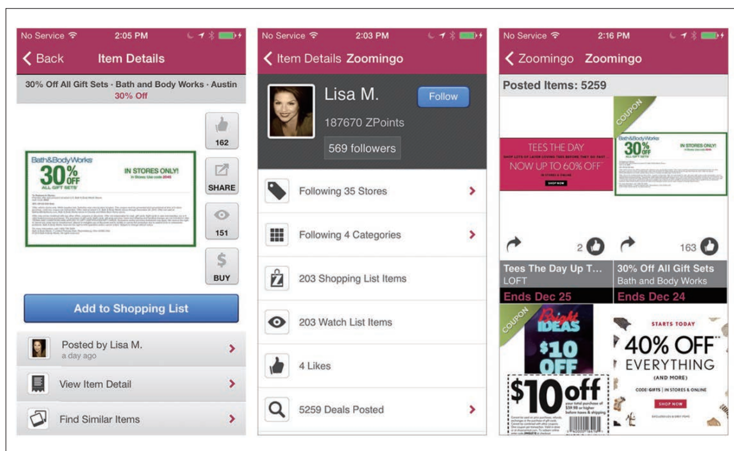



图 8-8

iOS 版 Zoomingo：“关注”流程突出展示了一个用户在该网络内的影响

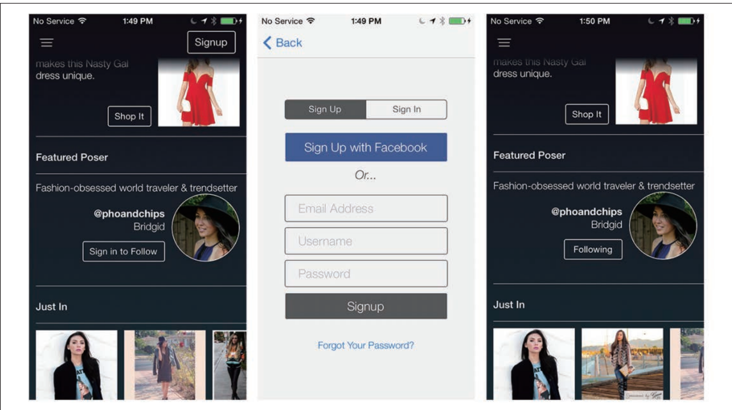
Pose 把“关注”变成了一个促使用户注册的功能。在它的“关注”流程中，首页显示一位推荐用户（Featured Poser）以及一个行为召唤的关注按钮。如果我想看这位推荐用户的收藏集和个人资料，就需要注册（如果我还没有注册的话）。

［小贴士］

在浏览流程的关键点上提供“关注”功能以增加用户的互动。要让个人资料页面上的“关注”按钮起到明显的行为召唤作用。

图 8-9

iOS 版 Pose：推荐用户使“关注”成为一种用户转化策略



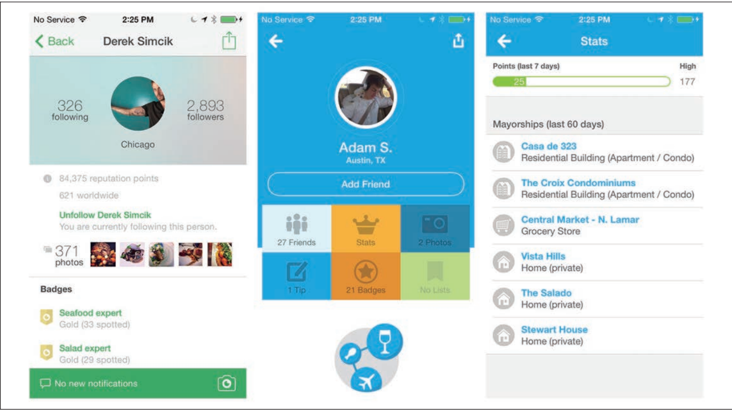

# 个人资料

虽然不是只有社交平台上才有个人资料，但在所有社交网络中，个人资料都是关键元素之一。最好的个人资料页会用照片、简介和兴趣列表“介绍”一个人。个人资料页还应该包括近期动态或其他说明该用户互动水平的指标，以及他的各项成就（受欢迎程度、影响力或在该网络中的活跃度），页面上还要提供关注、加为好友或是其他与那个人连接的功能。

一些应用使用统计数据显示一个人在该社区的互动性，比如 Foodspotting 和 Foursquare。这种设计合乎情理，因为这两个应用中的关系都是建立在信用和信任的基础上。

图 8-10

iOS 版 Foodspotting 和 Foursquare：用统计数据来表示互动和成就



Pose 和 Pinterest 以注册用户分享的兴趣为基础，是更偏向于视觉的社交网络。那么用户个人资料页的设计以分享照片内容为主也就不足为奇。

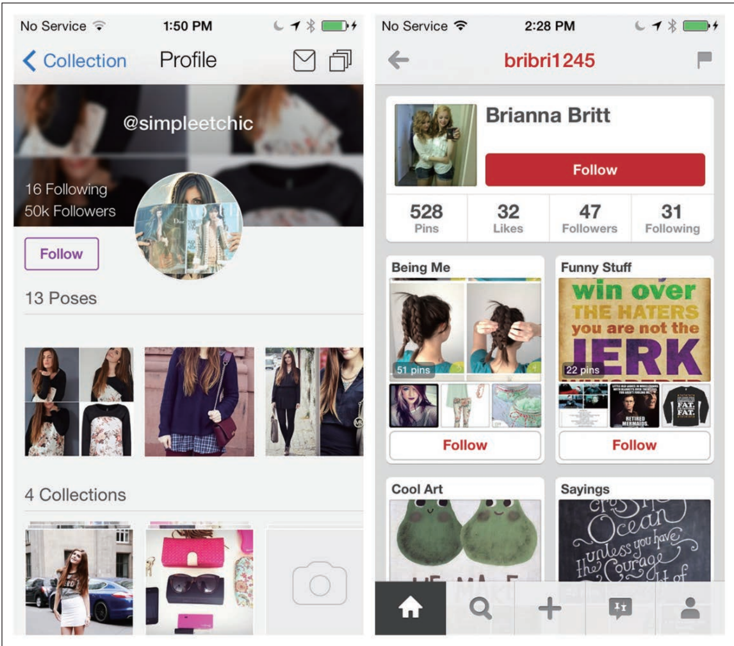



图 8-11

iOS 版 Pose 和 Pinterest：个人资料页上突出显示了所分享的内容

有时候，一个社交网络涉及个人生活的许多方面，以至于一页个人资料根本放不下那么多内容。Instagram 和 Google+ 采用选项卡呈现个人资料的不同方面，而 LinkedIn 则采用逐级深入的列表收录更多个人信息。

[小贴士]

个人资料是社交网络的核心。归根结底，其设计应该涵盖足够的信息，让大家愿意在这个网络中彼此连接，建立关系。

图 8-12

Instagram 和 Google+：选项卡式的个人资料页

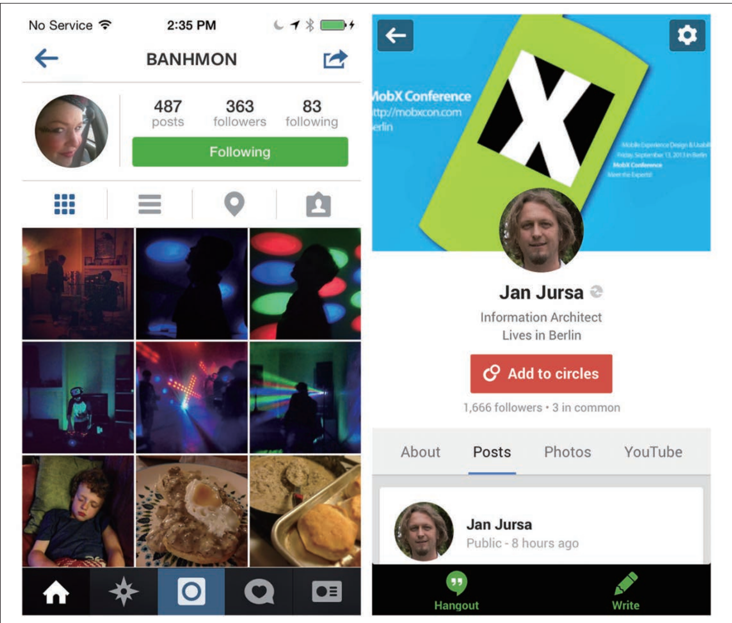
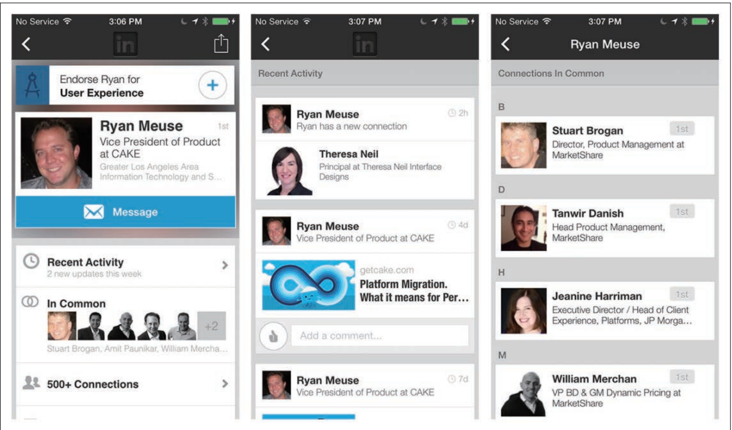



图 8-13

iOS 版 LinkedIn：更多个人资料详情采用列表式设计




## 群组

一些社交网络的群组是内部设置好的，还有一些群组需要由用户创建。Group.me 的创建群组界面就极其简洁。首先新建群组名称，然后利用动态搜索来添加联系人。

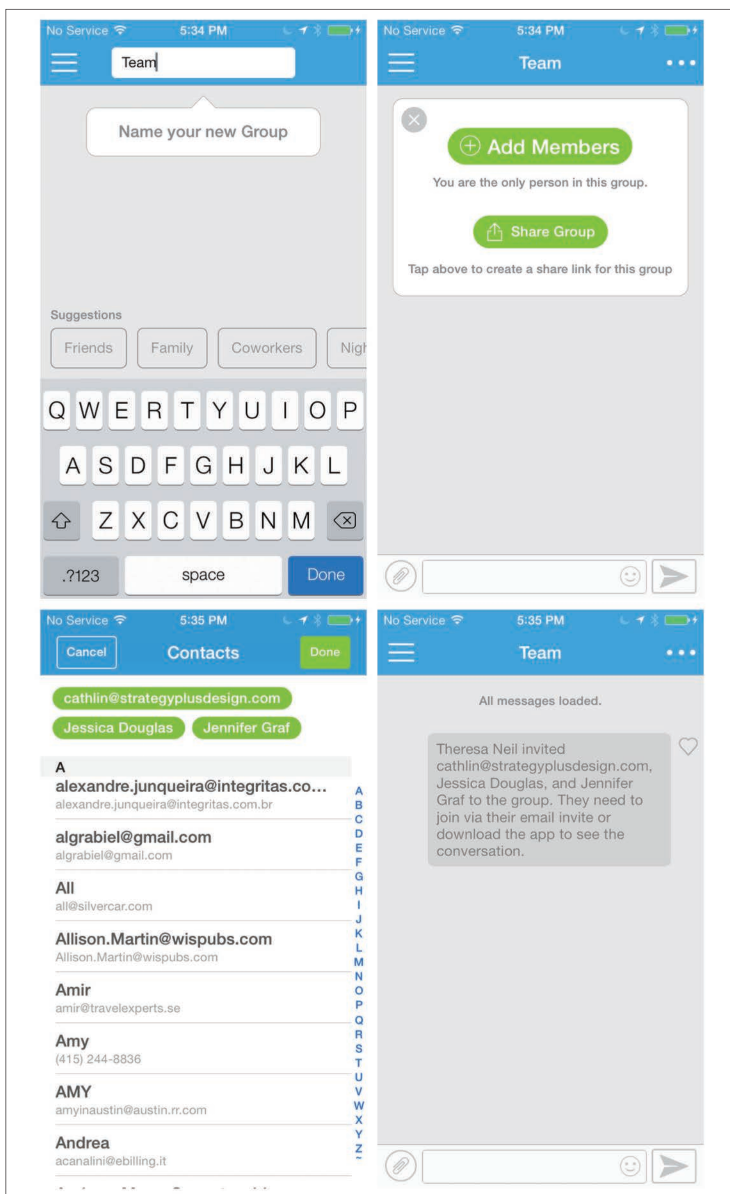


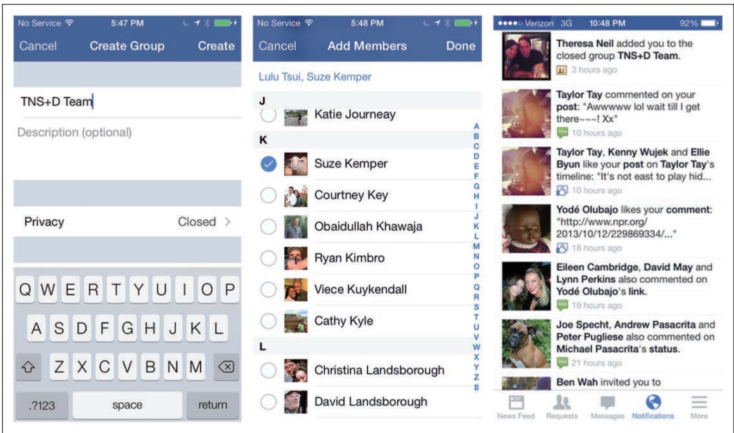

图 8-14

iOS 版 Group.me：利用动态搜索简化群组创建

在选择添加群组成员之后，系统会向他们发送邀请。在 Facebook 这个示例中，我收到的邀请显示为应用内提醒。

图 8-15

iOS 版 Facebook：创建群组并添加成员，被添加的成员会收到应用内提醒

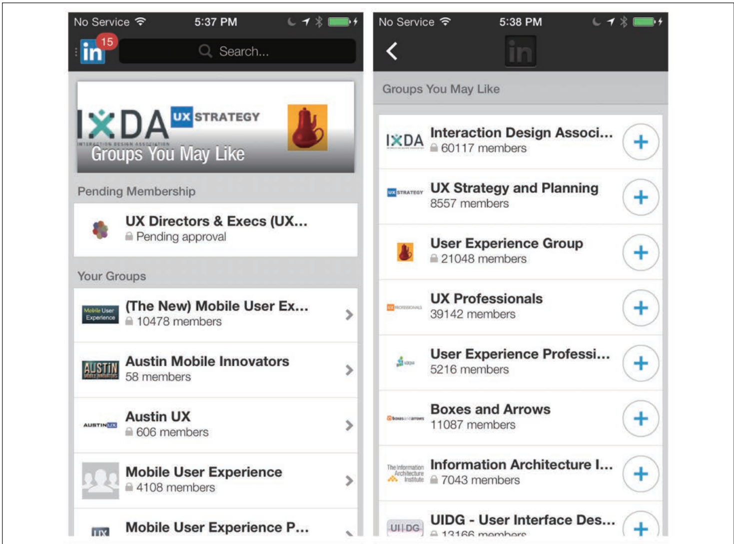



LinkedIn 会向你推荐“你可能喜欢的群组”，并允许你通过点击“+”号按钮加入该组。许多群组需要组长确认才可以加入。在此例中，我添加了 Boxes 和 Arrows Group 群组，然后得到反馈说我的请求正在处理中。我在我的个人资料页上面可以方便地访问我已经加入的 13 个群组。

如果你的社交网络确实需要用户创建群组，那么你应当把流程设计得极其简单，并在应用中使用视觉吸引。

图 8-16

iOS 版 LinkedIn：一个更正式地加入群组的程序



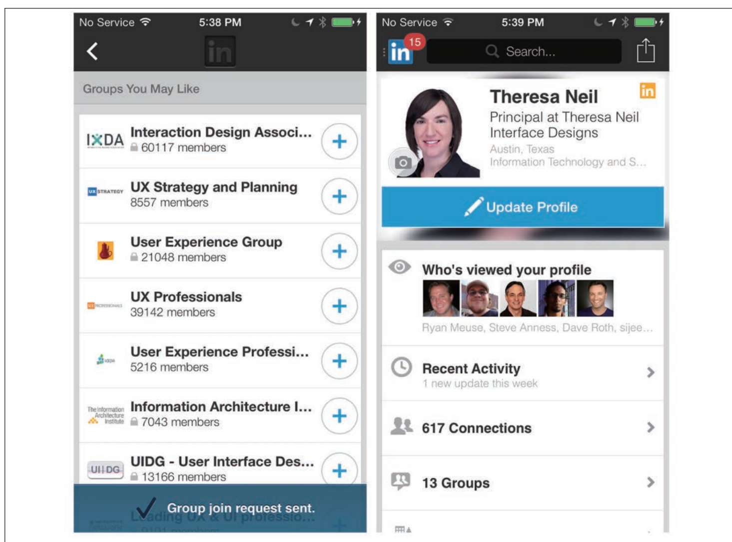



图 8-16（续）
iOS 版 LinkedIn：一个更正式地加入群组的程序

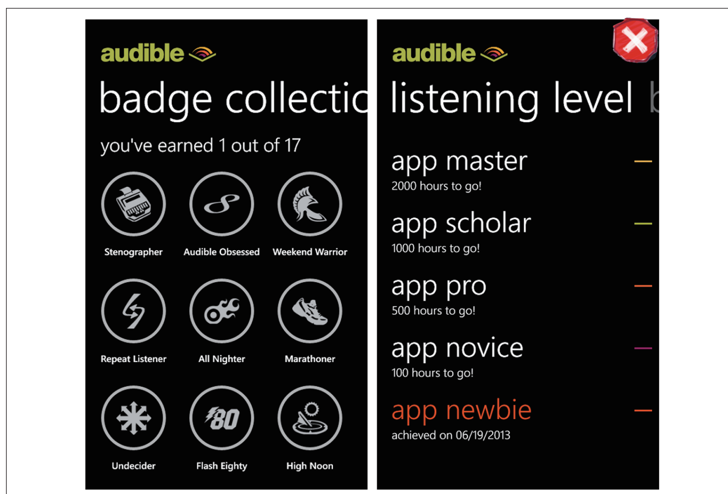
# 游戏化

如今，似乎人人都想把自己的应用游戏化（非游戏类应用中包含类游戏元素）。实际上他们真正想要的是一个有黏性的应用，即人们会反复使用的应用。但真正的游戏化远不止提供几个徽章那么简单。要想理解游戏化背后的科学和心理因素，我强烈推荐阅读以下三本书：《行为变化设计》（*Designing for Behavior Change*，Stephen Wendel 著，由 O’Reilly 于 2013 年出版）、《说服人要懂心理学》[1]（*How to Get People to Do Stuff*，Susan Weinshenk 著，由 New Riders 于 2013 年出版），以及《怦然心动：情感化交互设计指南》（*Seductive Interaction Design*，Stephen Anderson 著，由 New Riders 于 2011 年出版）。

请注意：要是游戏化实现得不好，可能会产生事与愿违的结果。让我们看看 Audible 以及我称之为“勿忘我”（Forget-Me-Not）的反模式示例。我是 Audible 的长期订阅用户，购买过几本部有声书籍。我在车里听 Audible，在电脑上听，在健身房里听，在各种不同的设备上听。虽然我是一个长期忠诚的用户，但看看我的收获吧：没得到任何徽章不说，它还把我列为了“新人”等级！

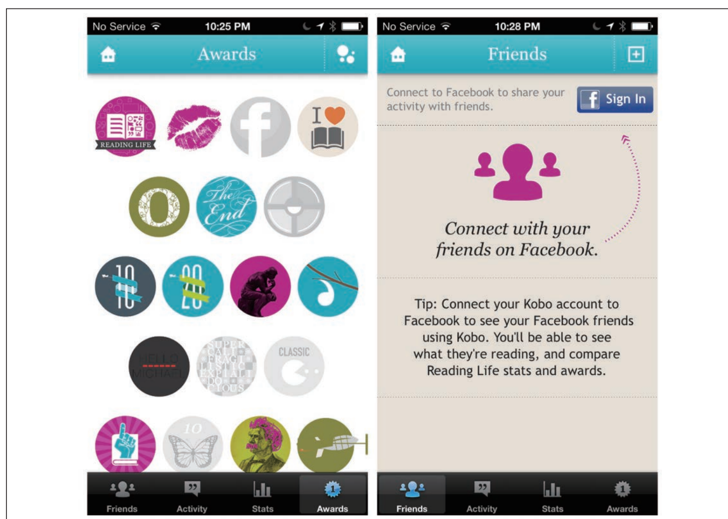
注 1：中文版已由人民邮电出版社出版。（编者注）

图 8-17
Windows Phone 版 Audible：虽然是长期的忠实用户，却依然是新人等级




为什么会这样？因为它只计算了我在某一个移动设备上所做的操作。要想实现游戏化，关键是要满足用户跨设备互联和同步的期望。Kobo 的 Reading Life 系统在这方面做得不错，而且还加入了“连接”设计模式。

图 8-18
iOS 版 Kobo：跨平台的 Reading Life 激励机制




耐克训练俱乐部（Nike Training Club）收回了徽章系统，但继续提供直接与个性化训练目标相关的激励机制。

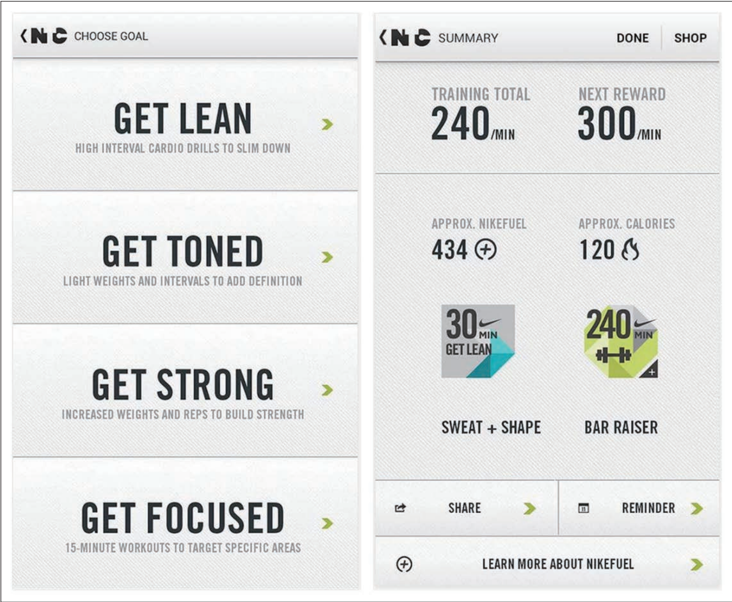



图 8-19
安卓版耐克训练俱乐部：实现个性化目标，然后获得奖励

Noom 利用游戏中的“升级”概念以鼓励用户不断参与和提高。它还采用了一种众所周知的简单易行的行为改变技巧：惊喜元素。我刚完成设置流程就得到了“优胜者奖金”，这激励我进一步参与进去。

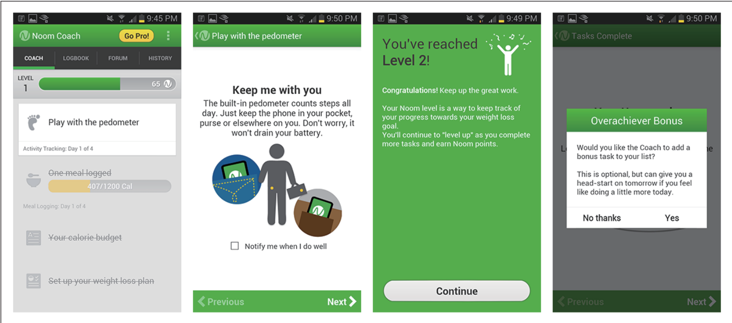
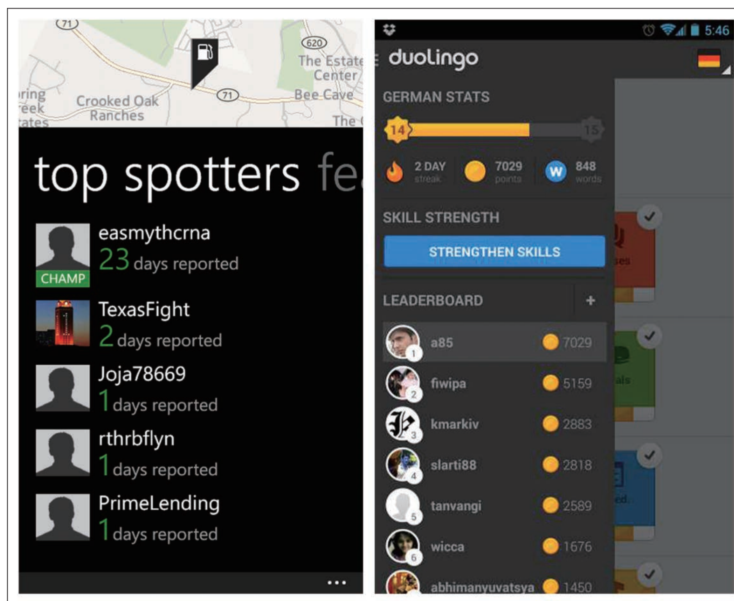



图 8-20
安卓版 Noom：升级和惊喜让体验游戏化

游戏化应用中的其他激励机制还包括排行榜（Leaderboards）和点数（Points）。GasBuddy 以“侦查员榜”（Top Spotters）为主要特色，而 Duolingo 则用数据展示最佳表现用户。

图 8-21
iOS 版 GasBuddy 和 Duolingo：排行榜和声望点数




［小贴士］

在尝试使用这种模式前，要理解游戏化和行为改变的原则。要保证模式简单有趣，不要在不适用的地方强行采用游戏化。

在 Duolingo 中，用户通过成功实现一组技能来挣得“宝石”（Lingots），然后可以用赚到的宝石购物，比如为可爱的小猫头鹰导师购买一套新的服装。

与 Duolingo 的简单机制相比，Foodspotting 的声望系统则显得冗长而混乱（http://www.foodspotting.com/about/faq），在其系统里，用户必须赚取点数，才能为喜欢的餐厅授予蓝色丝带，这个丝带叫作“吃货”（noms）。如果你的激励机制不能用一两句话解释清楚，就说明它可能太过复杂，以致不能如你所愿起到激励用户的作用。

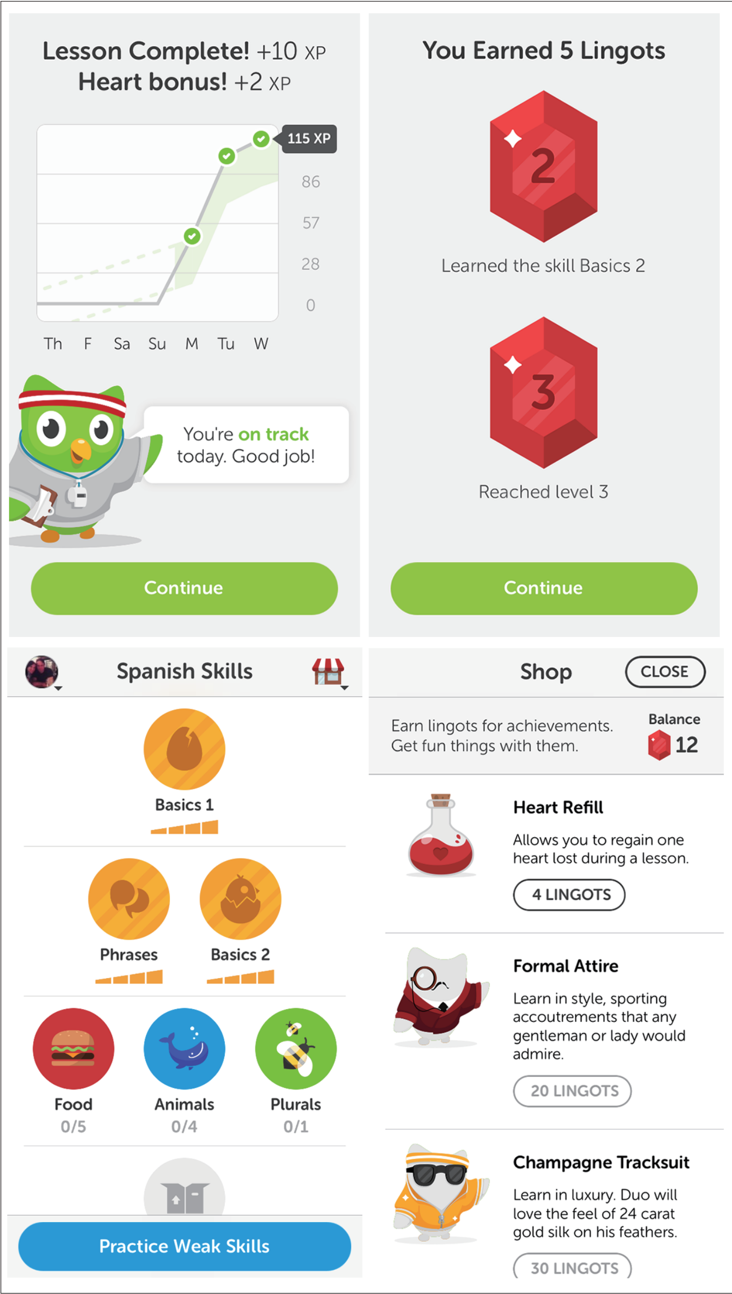



图 8-22

Duolingo 显示相对目标的学习进度，在用户完成课程后给予宝石奖励

第 9 章

# 反馈与功能可见性




- 反馈模式

  错误提示（Error Messages）、确认（Confirmation）、系统状态（System Status）

- 功能可见性模式

  点击（Tap）、滑动 / 轻扫（Swipe/Flick）、拖拽（Drag）

## 反馈模式

用户体验行业协会（User Experience Professionals Association，UXPA）在“可用性设计原则”（Principles for Usable Design，http://www.usability bok.org/principles-for-usable-design）一文中谈到，优秀的用户界面设计应该为用户“提供恰当的、清晰的、及时的反馈，好让他们看到自己操作的结果，并知道系统发生的变化”。我想借用一部老电影中

的经典台词，并简写成："给予反馈意味着用户永远不必说：'发生了什么事？！'"

反馈有各种形式，从简单的进度指示器和确认信息，到更为复杂的动画和特效。移动反馈模式包括错误提示、确认和系统状态。

## 错误提示

错误提示应该用直白而非代码式的语言表达出来，精确地指出问题所在，并给出建设性的解决建议。最佳实践是在一页上突出显示错误提示，这样的例子有 Soundwave 和 Pandora。这种设计方案比模态对话框更好，因为模态对话框遮挡住了问题，比如 Square Register 和 Pinterest 的设计。

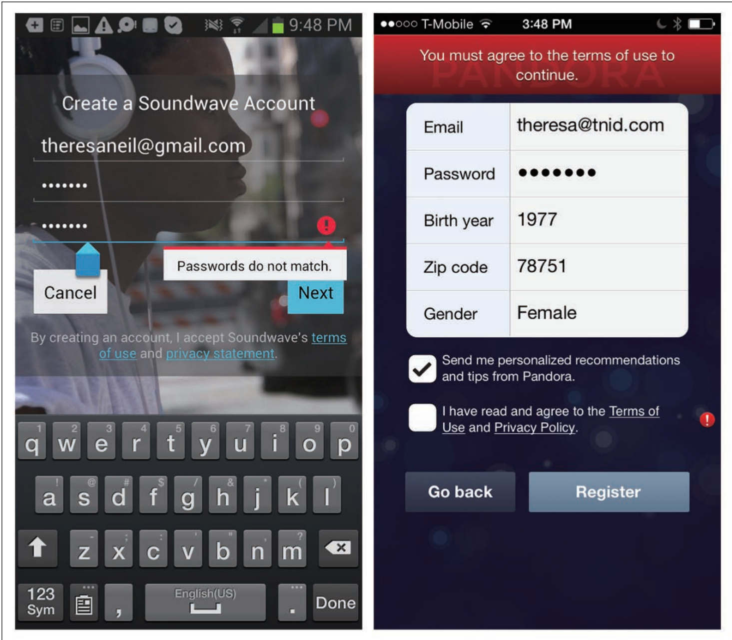



图 9-1
安卓版 Soundwave 和 iOS 版 Pandora：突出的错误提示

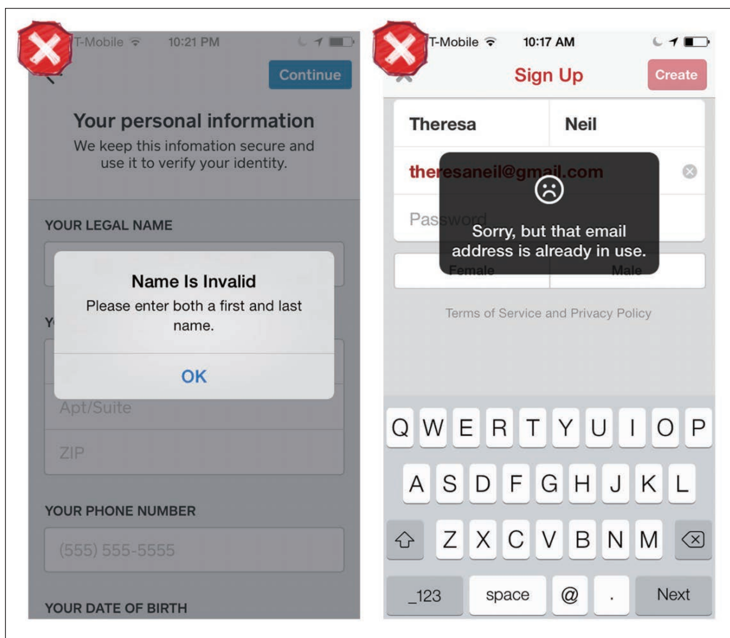



图 9-2

iOS 版 Square Register 和 Pinterest：错误对话框挡住了错误

如果你能预防错误，那就更好了。参看“确认”模式中对预防错误的行内反馈及错误提示的设计建议。

## 确认

在执行一项操作时，要提供确认。但不要使用“愚蠢的对话框”反模式，比如安卓版亚马逊和 HauteLook 应用，而要寻找不会打断用户操作流的提供确认反馈的方法。(更多关于“愚蠢的对话框”的反模式的内容，请参见第 11 章“愚蠢的对话框”一节。)举例来说，iOS 版亚马逊和 Zappos 应用都使用了动画来表示一件商品被加入了购物车，且购物车的商品数量随之更新。

［小贴士］

用直白的语言说明错误，并提供解决方案。把错误提示设计在页面内，好让错误可见，而不是把错误遮盖住。

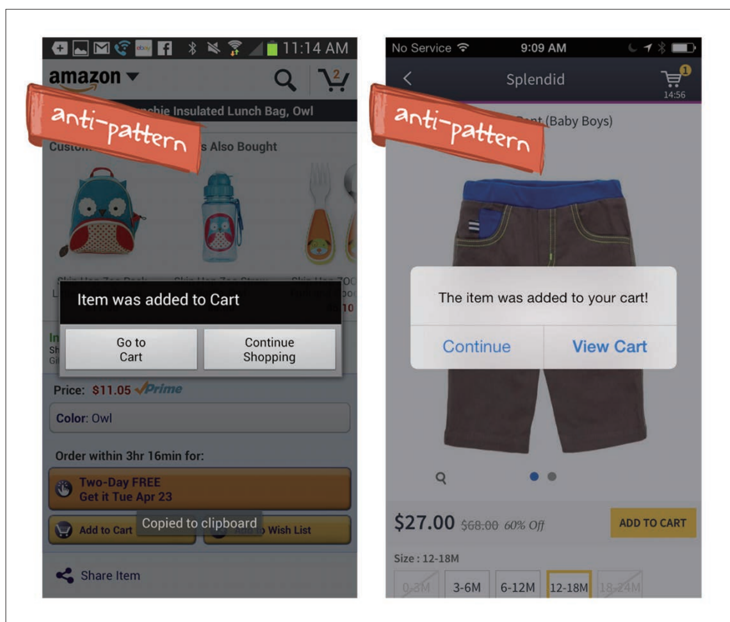



图 9-3
安卓版亚马逊和 iOS 版 HauteLook：愚蠢的确认对话框打断了购物流程

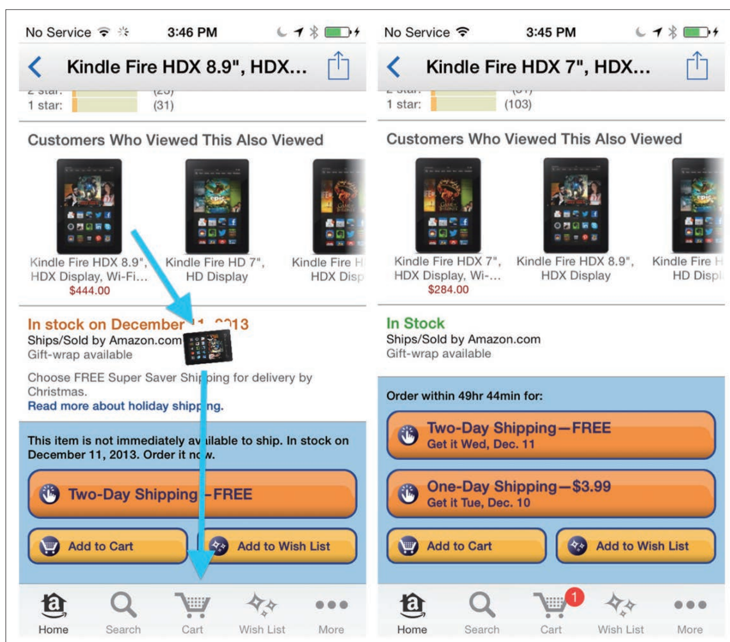
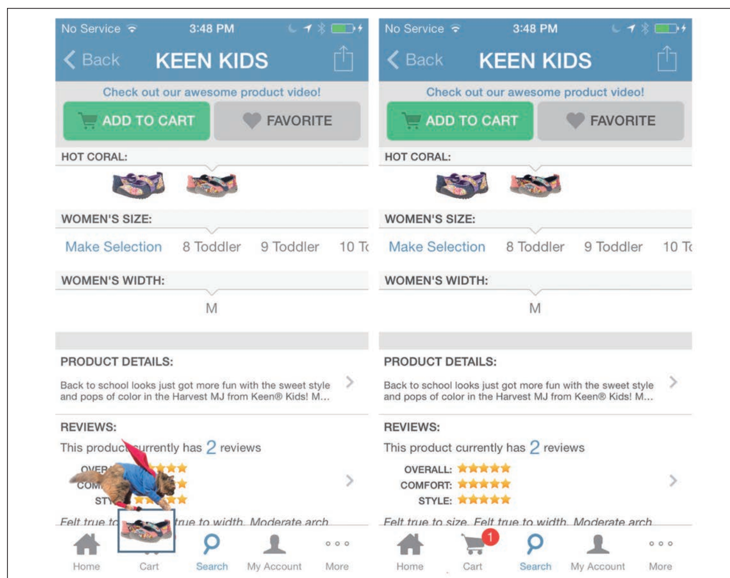



图 9-4
iOS 版亚马逊：Kindle“掉”到购物车里，购物车中的商品数立即更新（http://www.youtube.com/watch?v=--JzNARO_eM）

图 9-5

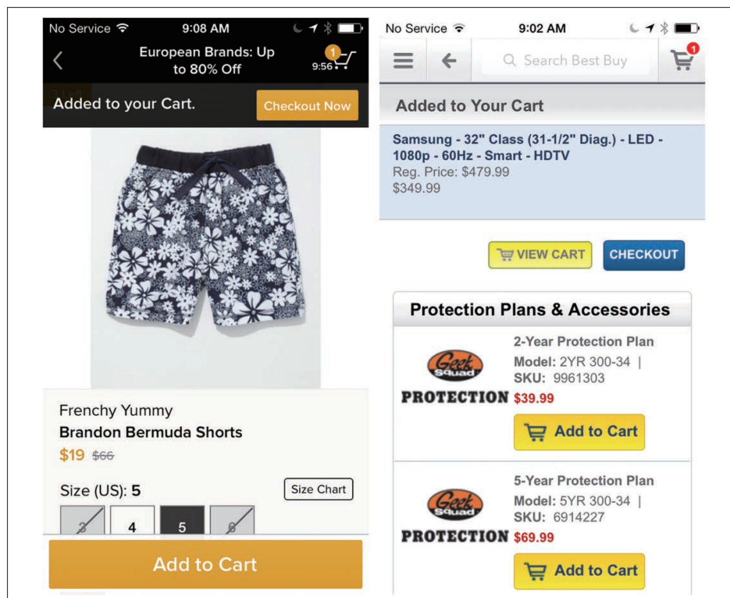
iOS 版 Zappos：超级猫“飞翔”着把一件商品送进购物车，购物车中的商品数立即更新（http://www.youtube.com/watch?v=JxADy-3HOjc）




另一种设计方案可见于 Gilt 和 BestBuy，这种方案除了更新购物车中的商品数，还会在购物车附近显示提示信息。

图 9-6

iOS 版 Gilt 和 BestBuy：在购物车旁边显示页面内提示，确认商品已被加入购物车



这些是购物车的示例，但是将一个项目加入任何类型的列表中时都需要用户确认。比如说，RetailMeNot 使用了动画来显示一项内容被保存到什么地方去了，同样重要的是，显示了此后可以在哪里找到保存的项。

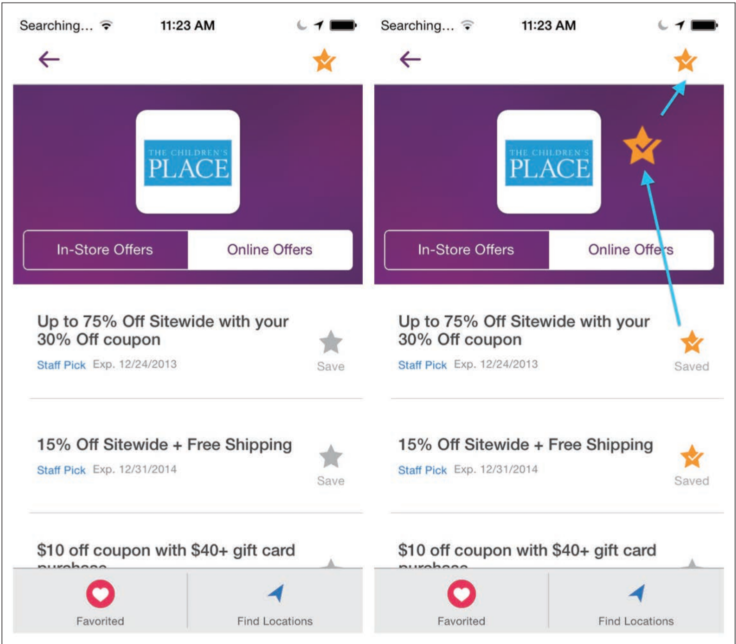



图 9-7
iOS 版 RetailMeNot：动画显示保存操作以及在哪里可以访问保存的项（http://www.youtube.com/watch?v=wFqxj1o4ueE）

行内反馈是一种很好的提供确认的设计模式，这样的例子有 iOS 版 Kik。这种模式特别适用于表单，当提交无效数据时，会触发多个错误提示。

行内反馈也应该用于功能性的手势，比如滑动删除。如果你不熟悉安卓滑动删除这个设计惯例的话，我要提醒一下，这个操作手势从根本上不同于 iOS 上的滑动删除。

我们以安卓版 Gmail 为例，滑动一封邮件会立即将其存档，这是典型的“滑动执行操作”（swipe-to-perform）的手势。行内反馈会显示这封邮件被存档了，同时提供撤销存档的选项。

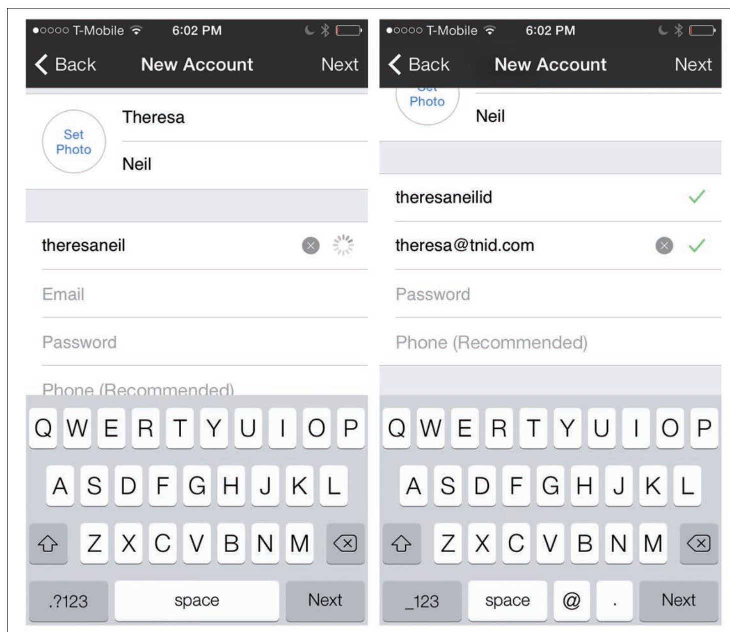



图 9-8

iOS 版 Kik：在填写表单字段时，行内反馈用来确认有效数据

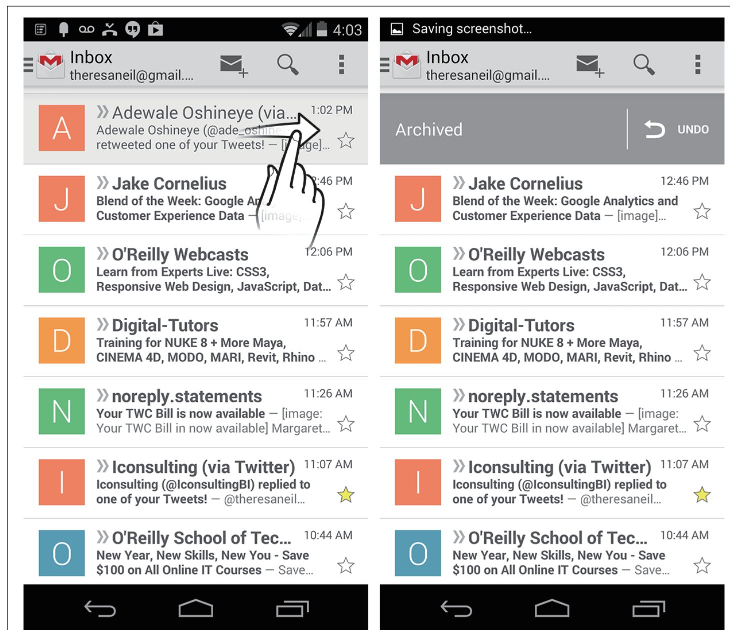



图 9-9

安卓版 Gmail：滑动删除手势触发行内确认，同时带有撤销操作

对照来看，iOS 系统的邮件应用（Mail）是“滑动显示”（swipe-to-reveal）手势的示例。滑动一封邮件时只显示可选择的操作，而没有执行任何操作。在这个例子中，需要使用第二个手势——点击，才会存档邮件。在这里不需要点击后的行内确认反馈，因为该邮件会从列表中消失，下面的邮件以动画效果自动上移。

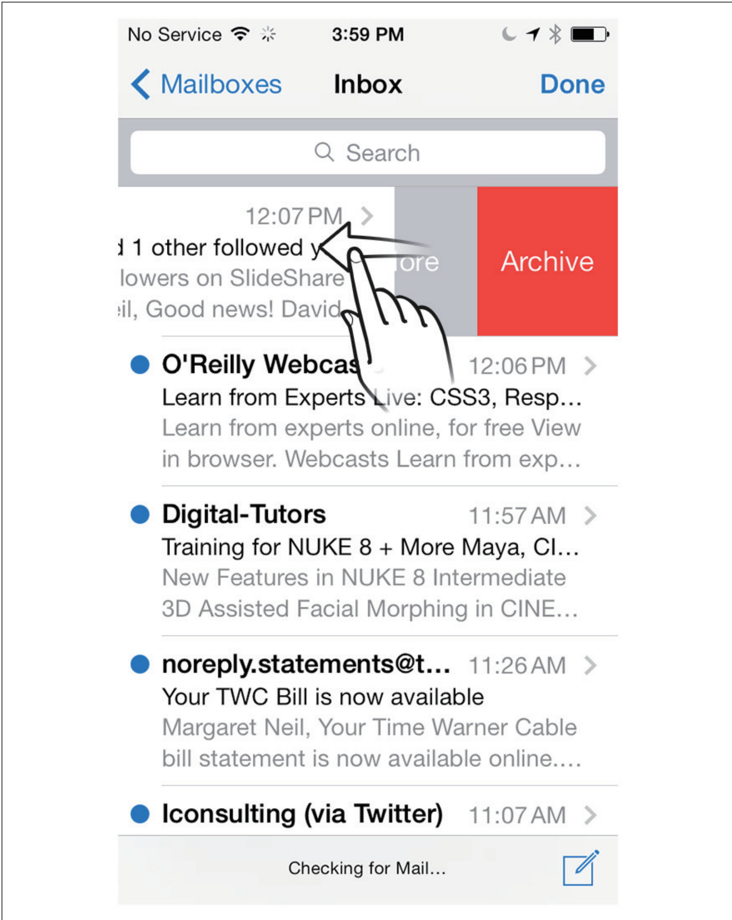



图 9-10

iOS 系统的邮件应用：滑动显示操作，点击存档；不需要行内确认

另一个提供确认的方案是使用一种叫作“Toast”的暂时通知。这种通知短暂地停留在屏幕上（5 秒或更短），然后消失——不需要用户进行额外的点击操作。Toast 通知确认刚刚执行的操作，并提供一个撤销选项，如 Gmail 和 Boxer 中设计的那样。

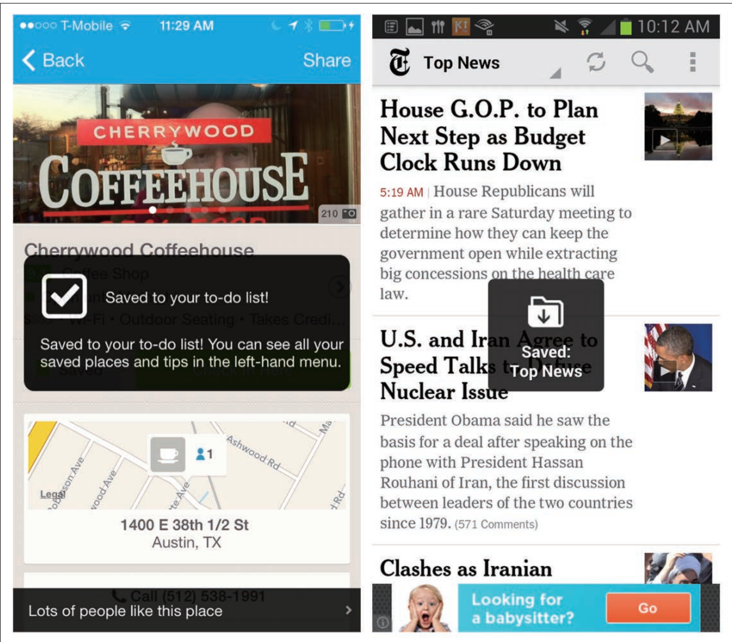



图 9-11

iOS 版 Foursquare 和 安 卓 版 NYTimes：简单的 Toast 通知

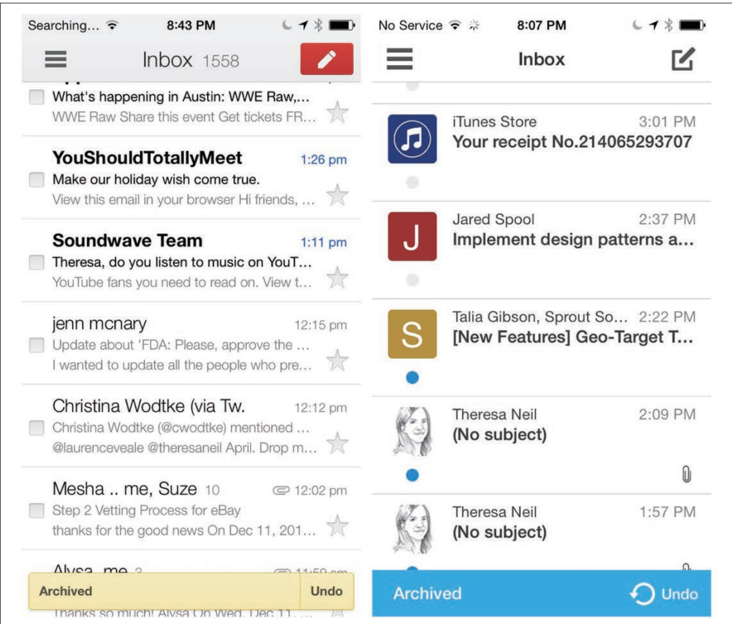



图 9-12

iOS 版 Gmail 和 Boxer：暂时 Toast 通知提供确认和撤销选项

像苹果 App Store 云端下载图标的多状态按钮和 Audible 指示下载进度的进度指示器，都是在不打断操作流的前提下提供确认反馈的方法。注意，Audible 在有声故事下载了一部分之后会显示提示“点击播放”，让你知道不用等到下载完就可以开始收听。

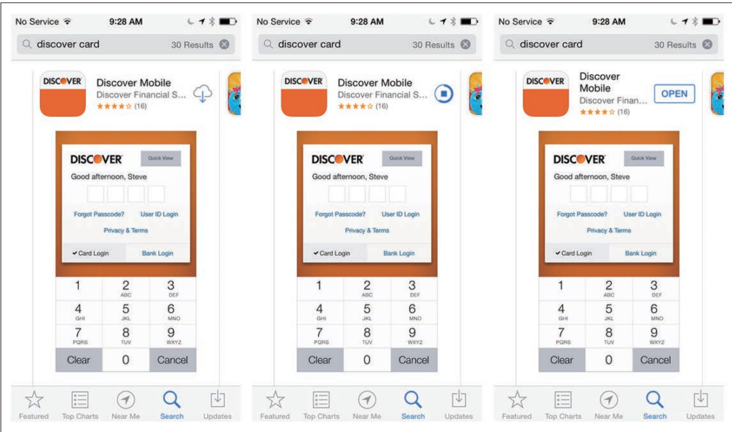



图 9-13

iOS 版 App Store：多状态按钮从“下载”变成“进度”指示，再变成“打开”

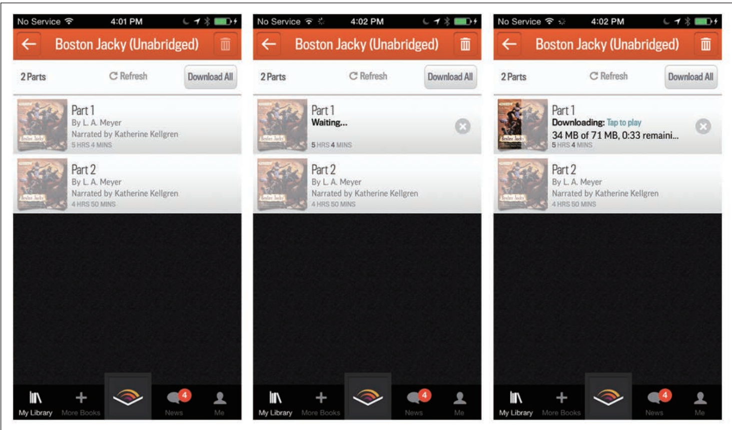



图 9-14

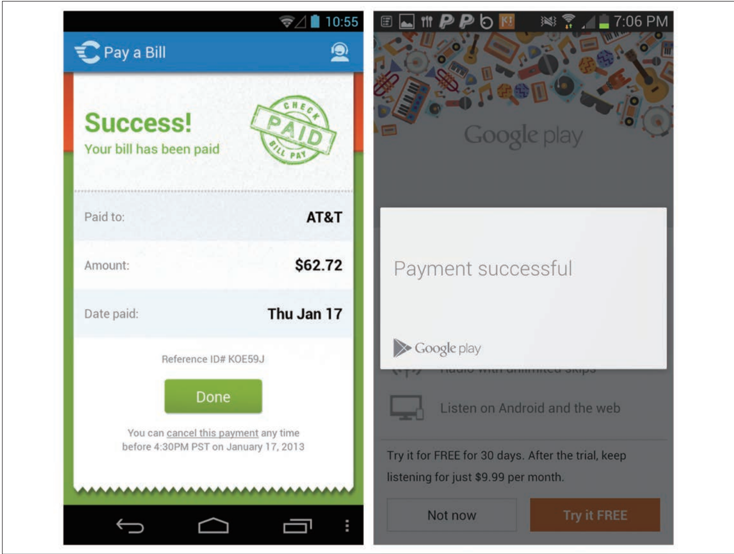
iOS 版 Audible：进度指示器提供反馈和行为召唤按钮

在完成一个事项后，比如下一个订单或提交一张表单之后，通常应该提供更突出的反馈。因为这些事件常出现在流程的最后，所以这时可以使用全屏或对话框式的确认反馈，像 Check 和 Google Play 中设计的那样。

[小贴士]

在执行操作后提供确认反馈，但不要打断操作流。把确认元素设计成整体流程的一部分。

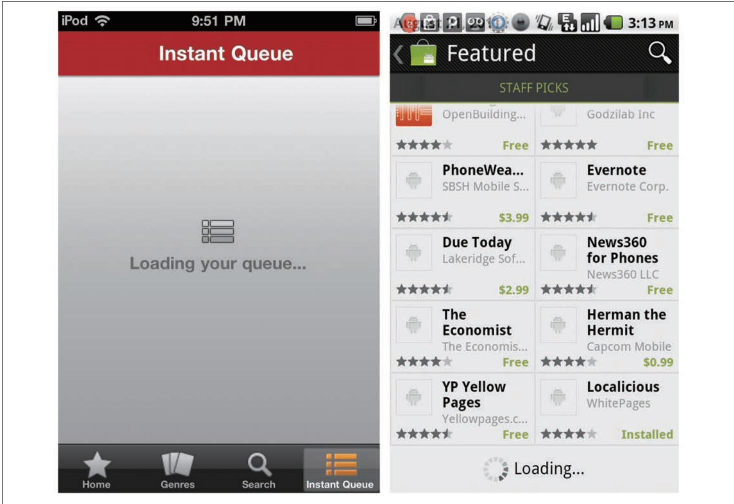
图 9-15
安卓版 Check 和 Google Play：完成一个事项需要给予醒目的反馈




## 系统状态

在本书第 1 版中，我鼓励移动设计师和开发者使用与网页端和桌面软件相同的技术来显示系统状态。通常来说，这些技术就是加载指示器。

图 9-16
本书第 1 版中的加载指示器示例



在那之后，第 2 章中我们提到 Luke Wroblewski 发表了一篇有趣的文章，题目是“避免使用 spinner 下拉菜单”（Avoid the Spinner，http://www.lukew.com/ff/entry.asp?1797=）。他开发的应用 Polar 发布了一个添加了 spinner 下拉菜单控件的版本，他说从那时起他开始收到越来越多这样的用户反馈：

> 好像刷新和加载页面都要等待很长时间，看起来没有上一版那么快了。

如果 Luke 因为一个简单的 spinner 控件而受到那样的抱怨，那我只能推测，Level 应用在加载页面时显示一只缓缓爬过页面的蜗牛的设计，也会引起同样的抱怨。

Luke 总结道：

> 我们使用的这些进度指示器促使用户盯着加载画面看，就像看着表针一分一秒走过去一样。结果人们觉得时间变慢了，整个应用的速度也变慢了。我们专注于指示器而不是进程，由此得出的结论是，你应该清楚地让用户明白，你在向目标推进，而不是让用户等待加载画面。

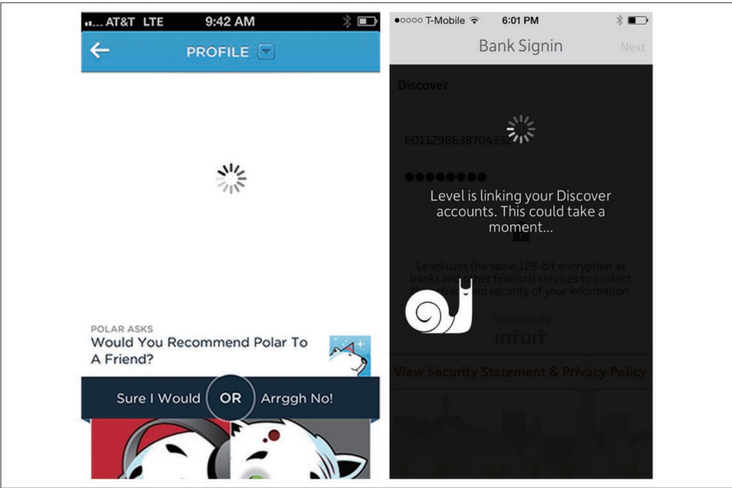
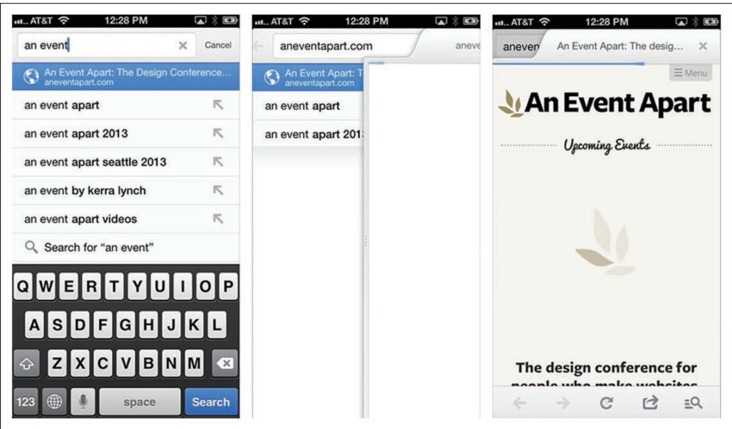



图 9-17
iOS 版 Polar 和 Level：让用户盯着加载画面

也有一些好的设计方案让用户专注于进程，而不是加载画面。比如谷歌的加载页面从侧边滑出，让用户感觉内容会立刻加载进来，即使情况并非如此。这种过渡动画实际上就是加载指示器。

图 9-18

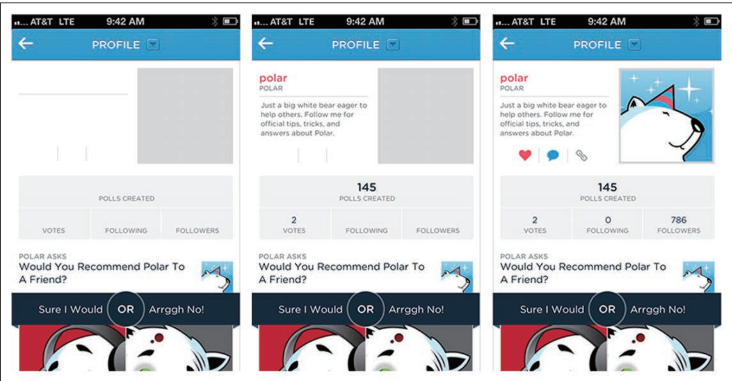
iOS 版谷歌应用：侧边滑出的过渡动画作为页面的加载指示器




现在 Polar 采用了构架页面（skeleton screens）的方案，构架页面本质上是一个空白页面，信息在该页面上逐步加载进来。

图 9-19

iOS 版 Polar：逐步加载的构架页面作为进度反馈




一些应用在初始化配置或注册时需要很长的等待时间。SigFig 和 Smartr 显示了两种让用户知道应用进程的方案。

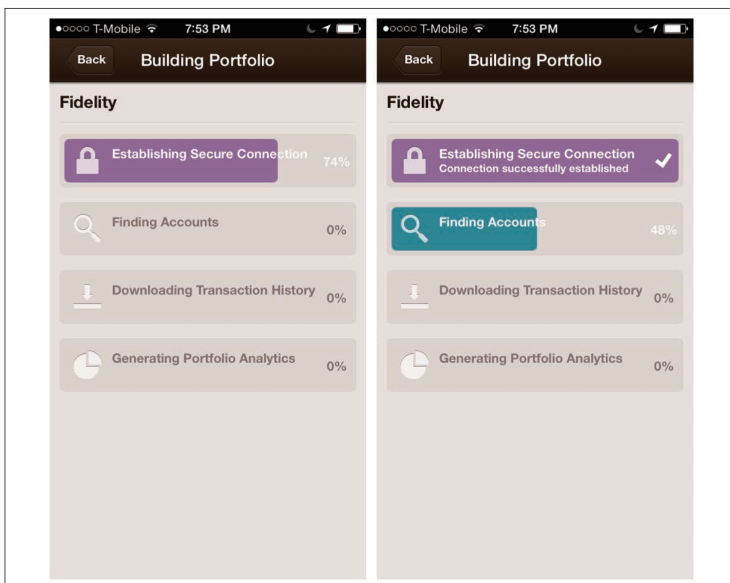



图 9-20
iOS 版 SigFig：在后台创建投资组合时，实时反馈显示每项任务的完成百分比

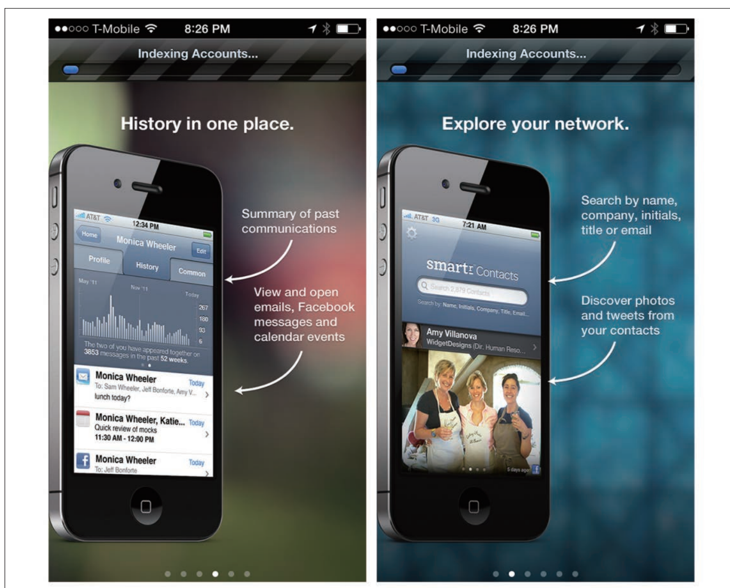



图 9-21
iOS 版 Smartr：在索引联系人的时候（进度条显示在上方）介绍产品功能

如果是很长的进程，要提供一个取消选项，比如 Story Board 在生成一部电影时的设计。

［小贴士］

为系统状态提供反馈，但要让用户专注于进程，而不是等待。为可能耗费时间的操作提供取消选项。

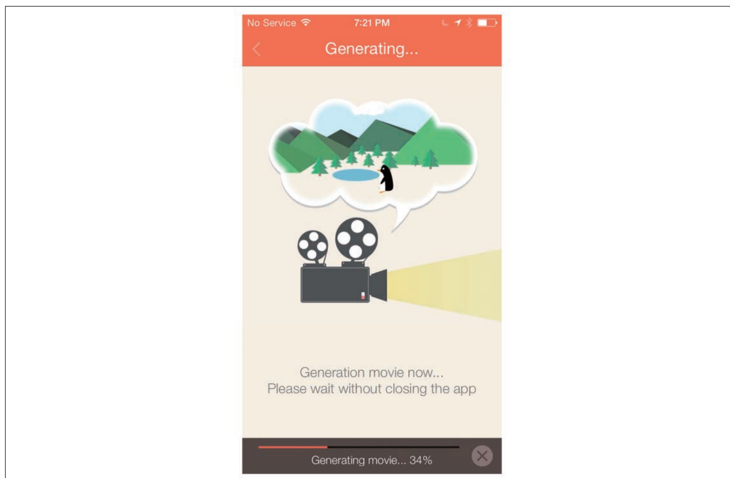



图 9-22
iOS 版 Story Board：长进程可以取消

## 功能可见性

功能可见性是指，一个对象具有让用户知道它可以用来执行一项操作的特性。抽屉把手和门把手就是现实生活中的示例。软件中常见的功能可见性示例包括拖拽手柄、卷起的页脚，以及按钮和滑块等 3D 控件。

由于移动设备可以利用单点触摸和多点触摸手势，可行的功能可见性方案很多，实现它们的技巧也很多。在此我们只讨论一些常见的点击、滑动 / 轻扫和拖拽方式的功能可见性。

### 点击

像斜面和阴影这样的视觉设计技巧可以让元素看上去是可点击的。然而，扁平化设计的兴起让这些技巧对设计师来说不再那么有效。实际上，一些人认为扁平化设计降低了许多应用的可用性。

Jakob Nielsen 是一位知名的用户体验咨询师兼作家，他在 Windows 8 上进行了一个可用性测试（http://www.nngroup.com/articles/windows-8-disappointing-usability/）。他发现，操作系统的扁平化设计让用户难以分辨哪些东西是可以点击的。在下面的例子中，Nielsen 的许多测试题都错把“更改系统设置”作为图标下的一个静态标签，但实际上它是一个可点击的操作命令。

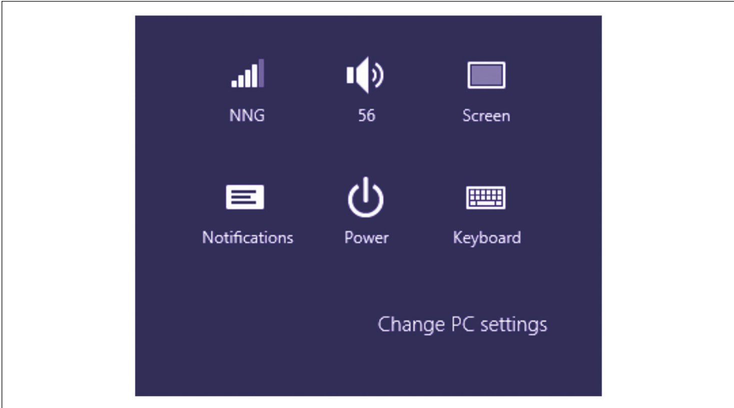



图 9-23
微软 Windows 8：扁平化设计可能降低了可用性

不是说扁平化设计不能体现功能可见性，只是它要求更高的设计技巧才能使其正确实现。Luke Wroblewski 在其撰写的另一篇精彩文章谈到了他在为 iOS 7 版 Polar 更新视觉设计时的经验（http://www.lukew.com/ff/entry.asp?1800），其中一个要点是：

> 要为页面上不同的元素设计出有意义的差异，去掉纹理和纵深使得剩余的视觉设计工作更加困难。我认为这是“扁平化设计”更难的一个主要原因。它迫使你简化，从而达到使用更少的视觉联系提供同样清晰的视觉传达的目的。

［小贴士］

差劲的扁平化设计会损害应用的整体可用性。在设计阶段准备好更多次迭代以保证可操作元素不会被扁平化视觉效果淹没。

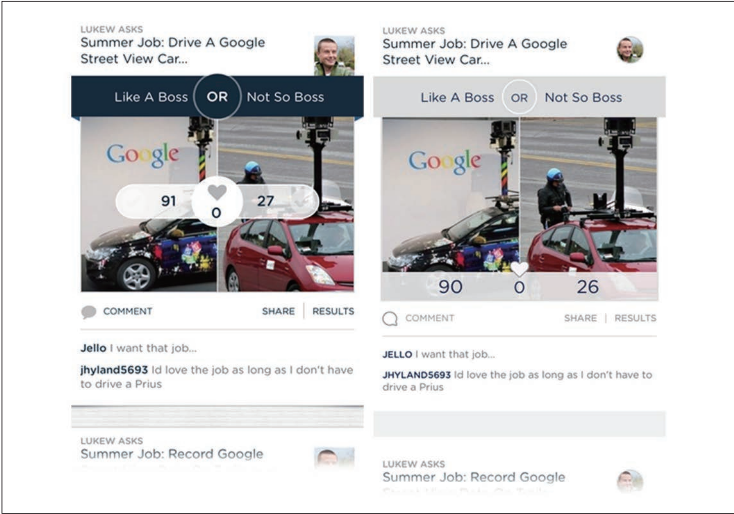



图 9-24
iOS 7 版 Polar 的早期设计迭代

## 滑动/轻扫

通过滑动操作实现功能可见性的方式有很多。类似 iOS 页面指示器控件（即页面底端的小圆点）的设计已被网页和移动应用所广泛采用。

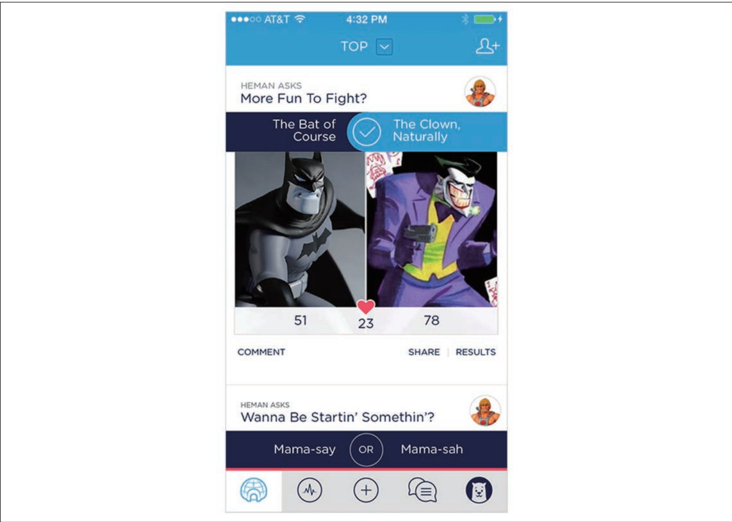



图 9-25

iOS 7 版 Polar：后来的迭代版本使不同的页面元素更明显

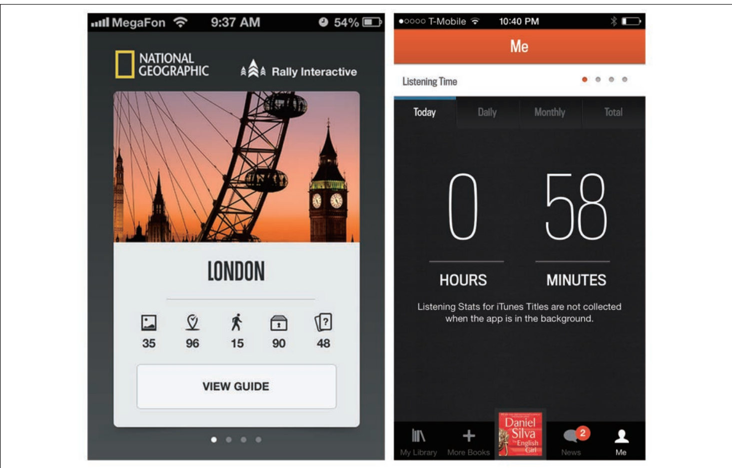



图 9-26

iOS 版国家地理城市指南（City Guides by National Geographic）和 Audible：页面指示器分别位于底端和顶端

还有一种方式是，在页面边缘不显示完整内容，以暗示页面侧边或下方还有更多内容。

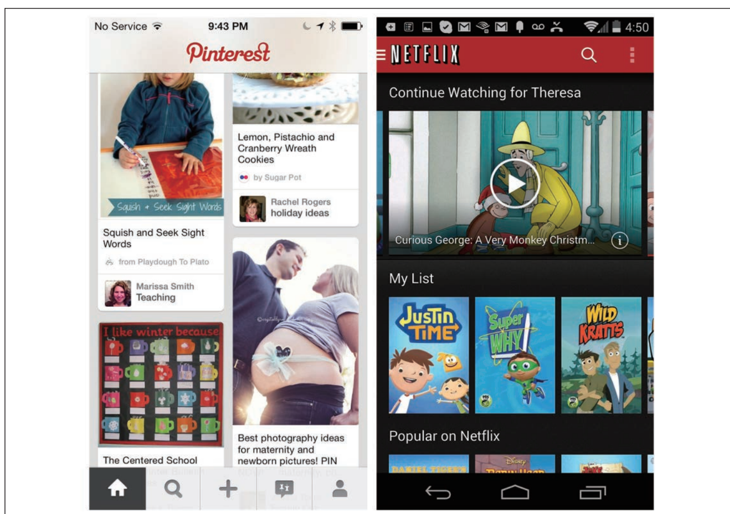



图 9-27

iOS 版 Pinterest 和安卓版 Netflix：页面溢出暗示如何操作才能查看未显示出来的内容

滚动条也是一种可选方案，尽管它在移动应用中不那么常见。滚动条也可用作快速跳转到某一位置的触摸对象。iOS 版 Skype 使用了字母表滚动条，让用户可以跳到特定的区块；安卓版 Skype 则使用了带有拖拽手柄的可隐藏式滚动条。

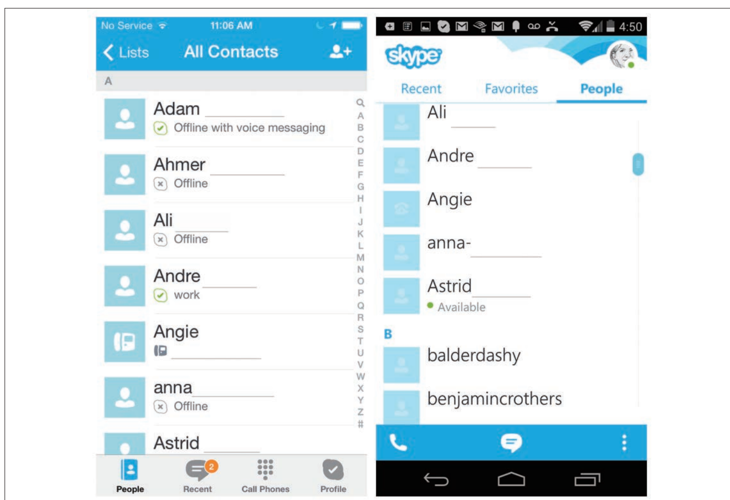



图 9-28

iOS 版和安卓版 Skype：不同的滚动条设计

其他为滑动提供视觉功能可见性的设计隐喻还包括封面流（Cover-flow）/ 轮盘、名片盒（Rolodex）以及堆叠的照片（Photo Stack）。

Potluck 使用了堆叠的卡片来引导滑动手势。向左滑动可以跳过当前的故事，向右滑动可以保存故事。然而在我第一次使用时并没有注意到这些，我不知道如果我把一则故事滑掉了，怎样才能找回它。之后我碰巧试着用拖拽手势代替滑动。拖拽会显示出一个提示，告诉我滑动会起到什么效果，但我不知道大多数人会先尝试拖拽还是先尝试滑动。

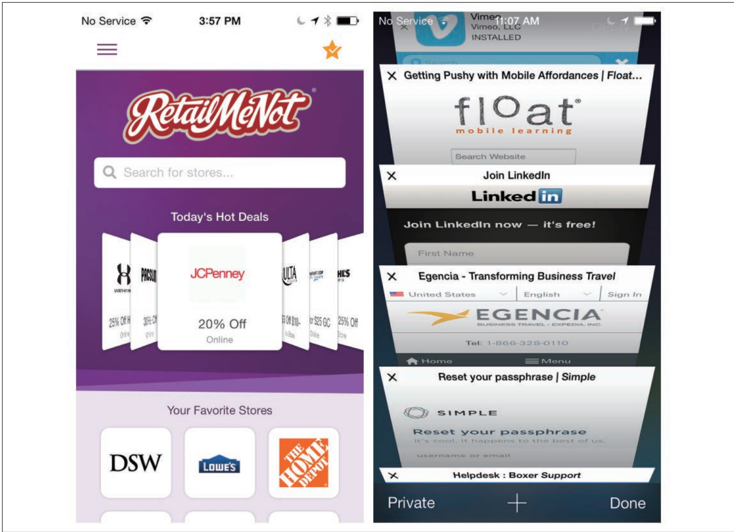



图 9-29

iOS 版 RetailMeNot 和 Safari：封面流的功能可见性设计暗示水平滑动，名片盒的功能可见性设计暗示垂直滑动

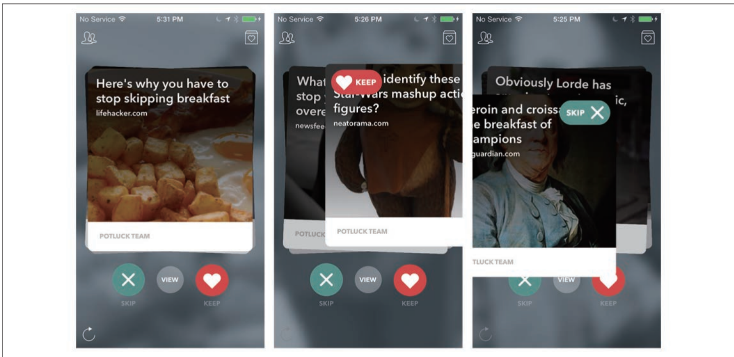



图 9-30

iOS 版 Potluck：优秀的滑动功能可见性设计，但对“滑走的故事到哪里去了”的提示可以设计得更　好（http://www.youtube.com/watch?v=pcfNFuvvdrA）

动画也可以吸引用户使用滑动手势。安卓版 Allthecooks 在侧边栏菜单打开时“弹跳”着提示：水平滑动手势可以显示完整菜单。Flipboard 上下翻动页面一角，引导用户使用其著名的垂直页面滑动手势。

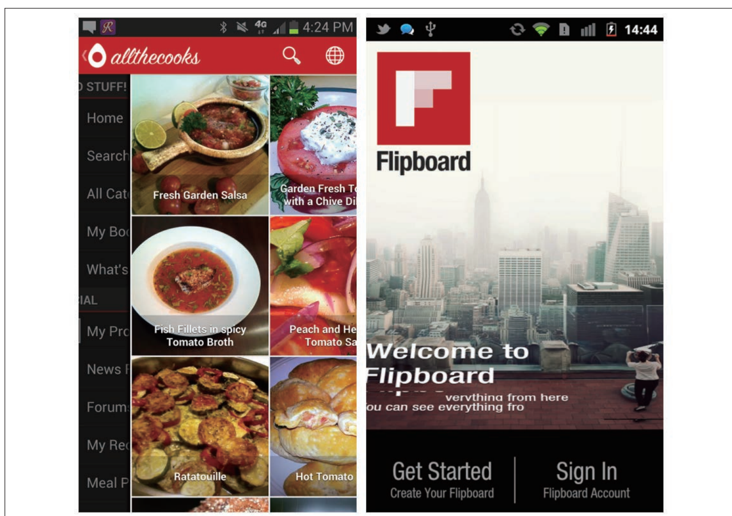



图 9-31

安卓版 Allthecooks（http://youtu.be/iL1Cs-gyNaE） 和 iOS 版 Flipboard（http://www.youtube.com/watch?v=rQASZtuvBCs）：用动画实现功能可见性

在第 7 章中，还谈到过 Mailbox 以及 Fantastical 的例子，说明了如何设计一个交互式导引，可向用户适时地介绍要用到的手势。

在吸引用户滑动操作方面，文字的方式起不到作用，这一点已被证实，所以不要采用这种方式，试试别的技巧吧。Silvercar 的早期版本用文字提示用户向侧边滑动，之后的版本已经改用垂直滑动。内容延展至屏幕底端外，提供了可以使用滑动手势的功能可见性。

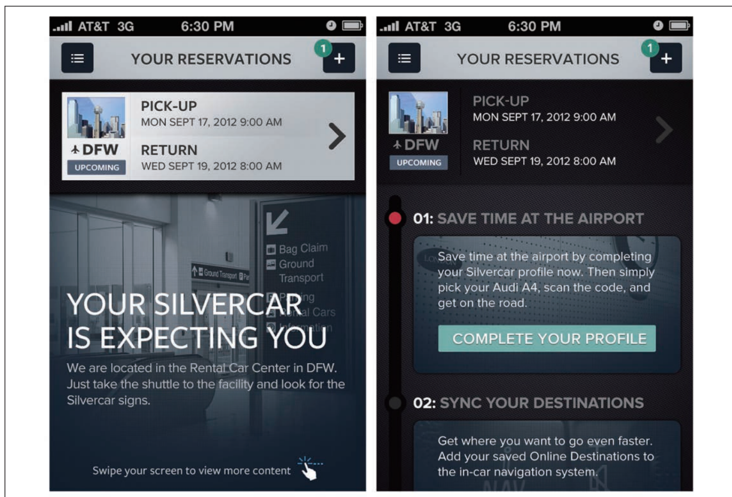



图 9-32

iOS 版 Silvercar：早期设计采用水平滑动和文字提示；改版后使用垂直滑动

## 拖拽

在移动设备上，一般先要进入一个模式才能使用拖拽操作，比如编辑模式，或者长按。在 iOS 中，你可以长按任何应用图标进入编辑模式，也可以拖拽应用重新排序。也可以在这个模式下删除应用。




图 9-33

iOS 7：长按可以拖拽应用图标或删除应用

在安卓系统中，长按一个应用图标会显示“拖放区域”，在这个状态下，可以把应用移动到新的屏幕或拖拽到卸载拖放区。像桌面端的垃圾桶图标一样，这一概念已经过时了。但这个设计中，真正反模式的地方在于，要想查看应用信息，用户必须先把应用图标拽进应用信息拖放区。就说说这有多么低效，多么繁琐吧。

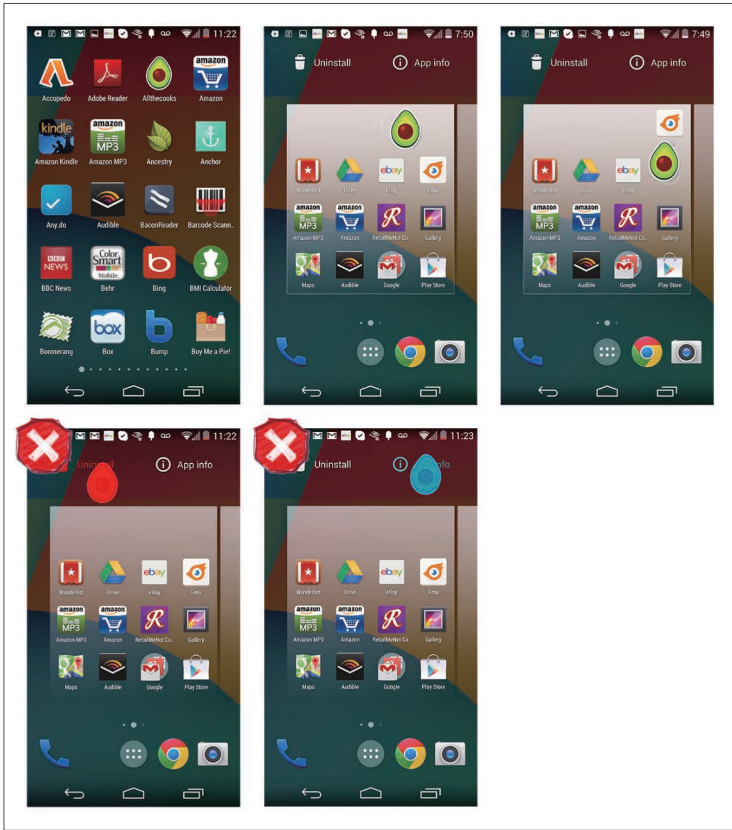



图 9-34

安卓操作系统：拖拽选定的应用到新的屏幕中，或是放到“卸载”或“应用信息”的拖放区

一个更好的解决方案是长按显示菜单，菜单的前几个选项是“移动”“删除”和“应用信息”。点击“移动”会激活编辑模式，并显示页面拖放区，就像现在的长按设计一样。

这些拖拽的示例全都没有任何功能可见性，用户必须自己去发掘。用户确实愿意把玩操作系统，学习一些小把戏，但不要断定他们也会如此对待你的应用。

在应用中，最常见的拖拽功能可见性是手柄。手柄常常在明确地进入编辑模式之后显示出来。手柄适合用来排序列表项，比如 Wunderlist 和 Lemon Wallet 中的设计。

图 9-35

iOS 版 Wunderlist：任务列表的编辑模式下，右侧显示拖拽手柄

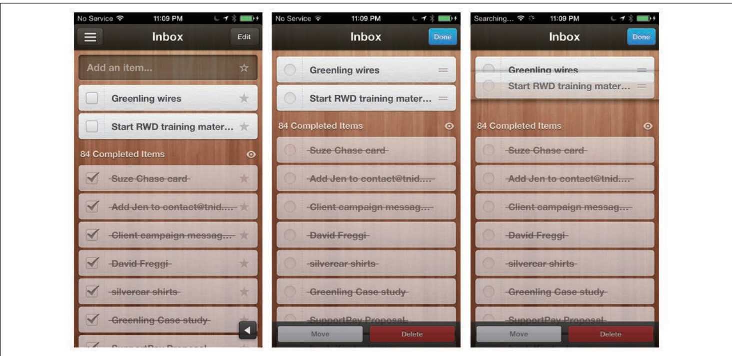
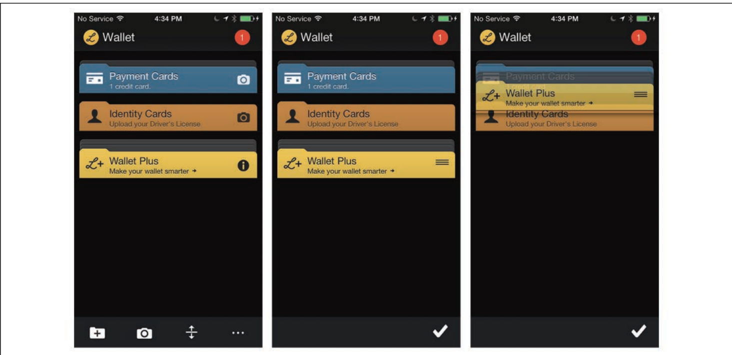



图 9-36

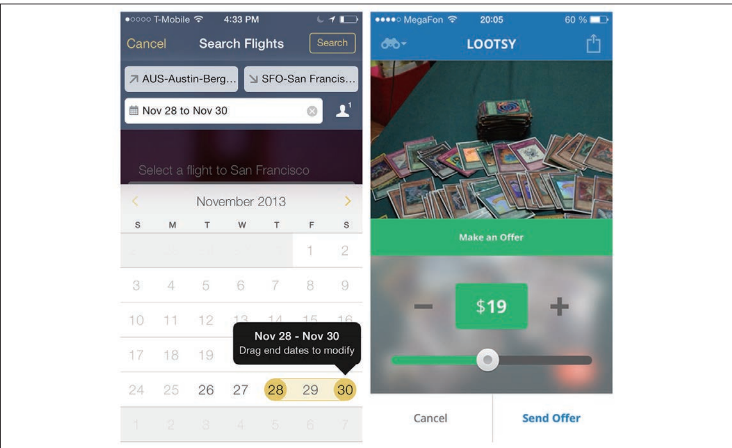
iOS 版 Lemon Wallet：点击操作栏的“拖拽”图标进入编辑模式，同时显示拖拽手柄




一些拖拽手柄默认可见，比如滑块控件上的手柄和抽屉中的手柄，如 Expedia 和 Lootsy 所示。

图 9-37

iOS 版 Expedia 和 Lootsy：一些拖拽手柄始终可见，比如滑块上的手柄



设计和配置类应用不只有拖拽移动，还拥有拖拽旋转和拖拽调整大小这样更复杂的拖拽交互。在拖拽功能可见性上，Aviary 和 Homestyler 都是值得研究的示例。

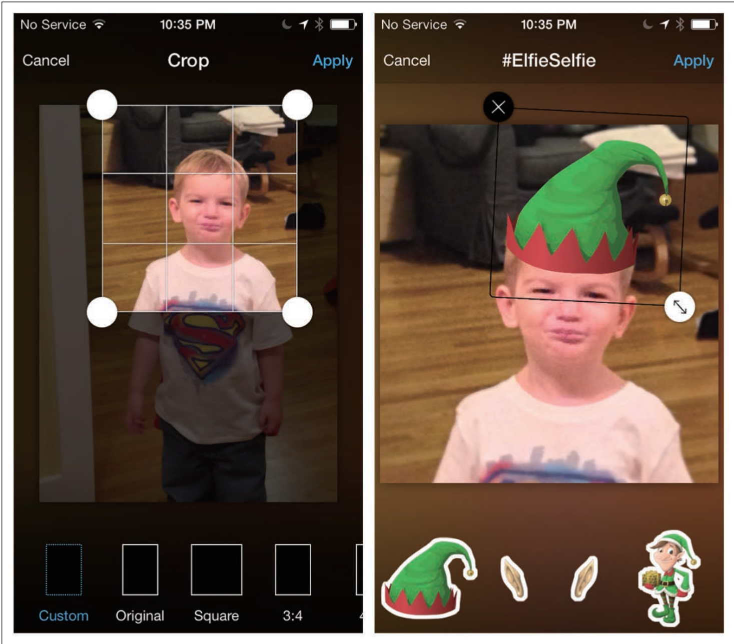



图 9-38
iOS 版 Aviary：多手柄裁切；单手柄调整大小、旋转以及放置贴图

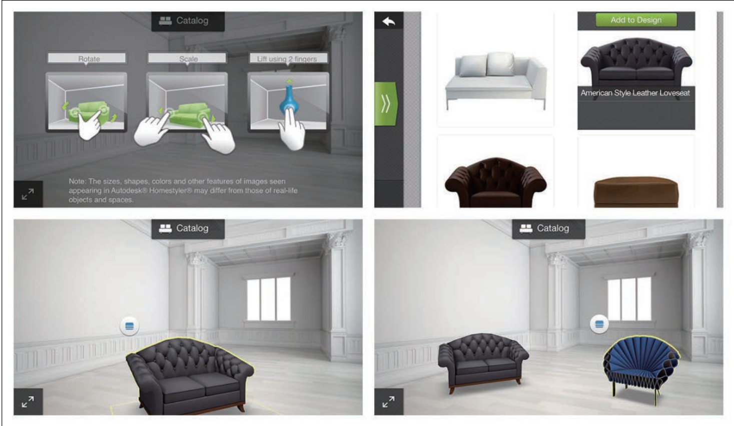



图 9-39
iOS 版 Homestyler：为重新布置所选物件提供了清晰的功能可见性

［小贴士］
拖拽手柄、选中对象和拖放区域要选用易于识别的设计方案。通过测试来判断是否需要设计一个视觉吸引向用户介绍该功能。

# 第 10 章

# 帮助




- 帮助模式

  入门介绍（How-Tos）、用户指南 / 帮助系统（User Guide/Help System）、常见问题解答（FAQs）、功能使用向导（Feature Tours）、新手引导（Tutorials）、上下文帮助（Contextual Help）、收集反馈（Capture Feedback）

我们认为优秀的移动应用应该很容易学习和快速掌握。但这不意味着应用设计师和开发者应该假设用户自己就能搞明白。

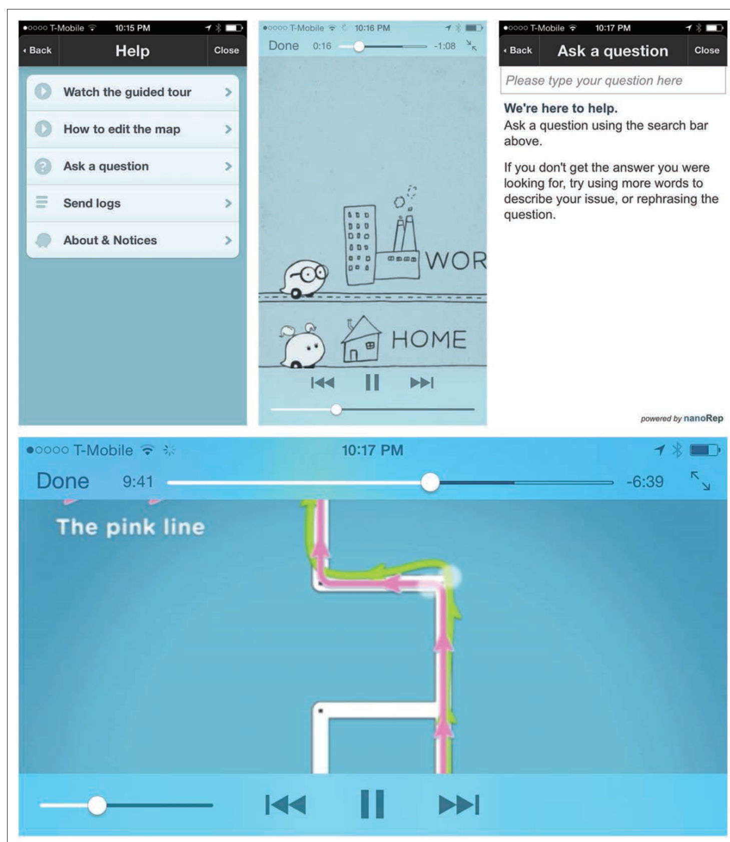
移动战略分析师 Michael Mace 在 VentureBeat.com 网站上发表了一篇文章（http://venturebeat.com/2013/09/21/90-of-mobile-apps-have-inadequate-help/），指出不应该做出如此随意的应用着陆设计方案。他说道："移动应用开发者应该假设用户会遇到困难，并为他们设计一条能够安然摆脱困境的路径。我们一次又一次看到用户努力想弄明白应该如何操作。即使是面对移动应用的线性功能，这样的情况也时有发生。"

当然，你了解你的应用，但那是因为你已经花费了好几周来设计、开发这个应用。不要以为对于用户来说，它们也是同样地直观。我们在此讨论的帮助模式可以让你的用户正常使用应用，或者至少能让他们摆脱困境：

- 入门介绍
- 用户指南 / 帮助系统
- 常见问题解答
- 功能使用向导
- 新手引导
- 上下文中的帮助
- 收集反馈

单一的帮助模式可能不够用。许多成功的应用都结合使用了视觉吸引和帮助模式来帮助用户发现功能，最大限度地利用好工具。举例来说，Waze 提供了一个欢迎向导、一个视频入门，以及一个交互式"问个问题"的常见问题解答。

图 10-1
iOS 版 Waze 结合多种模式提供应用内帮助




只要有可能，就应该把帮助内容直接嵌入到应用中，而不是重定向到外部网站。如果你必须重定向到一个外部站点，那就要确保其内容为移动设备做过优化，比如谷歌地图和 SnipSnap 应用。Kindle 和 Runtastic's Six Pack 应用就没能正确做到这一点，甚至给用户造成了更多困扰。

［小贴士］

直接在应用内嵌入帮助内容。如果要重定向到一个网站，就要确保内容为移动端做过优化。

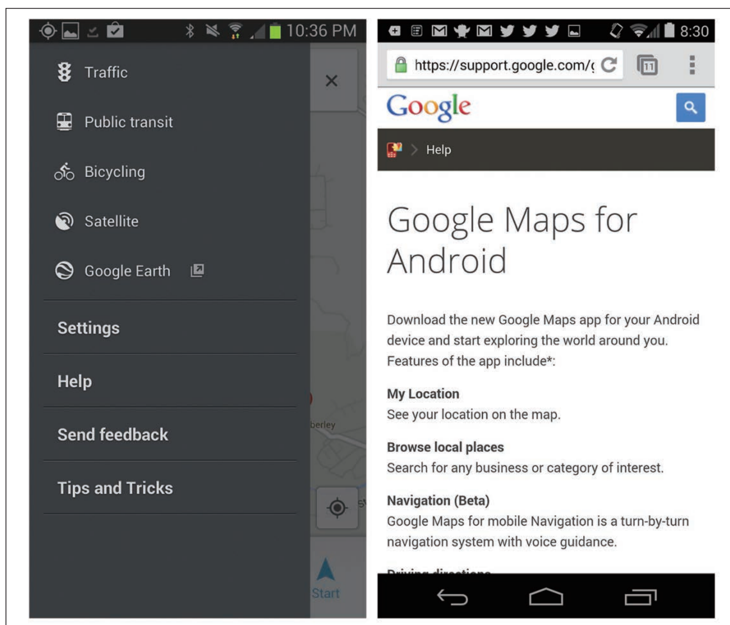



图 10-2
安卓版谷歌地图在浏览器中打开帮助

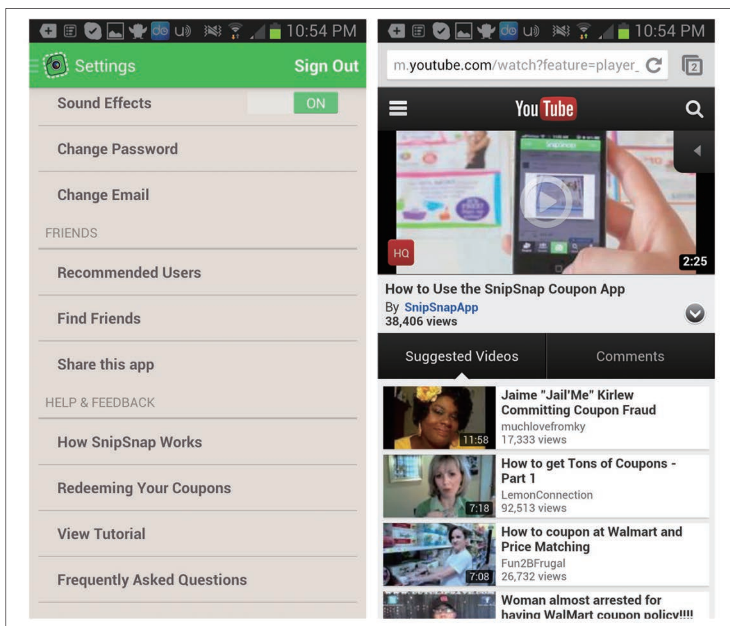



图 10-3
安卓版 SnipSnap 在浏览器中打开一个引导视频

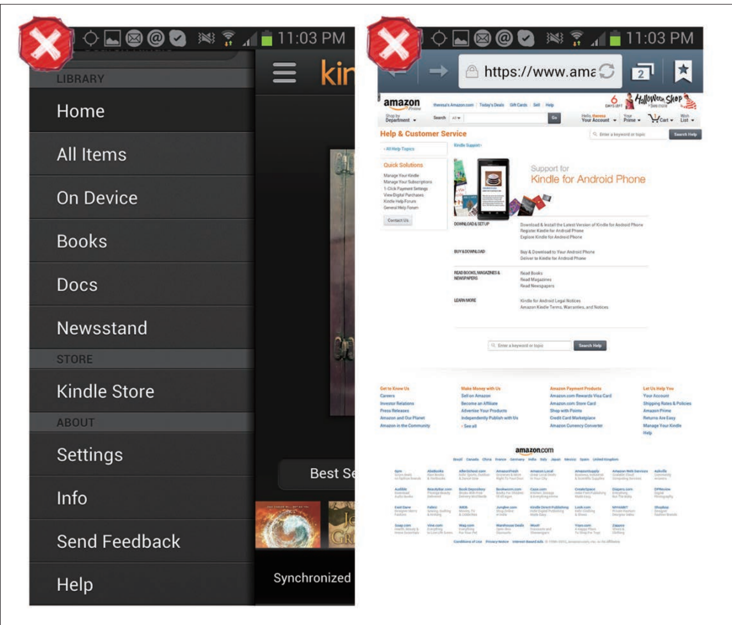



图 10-4
安卓版 Kindle 打开了桌面版网页，而不是移动优化过的网页版本

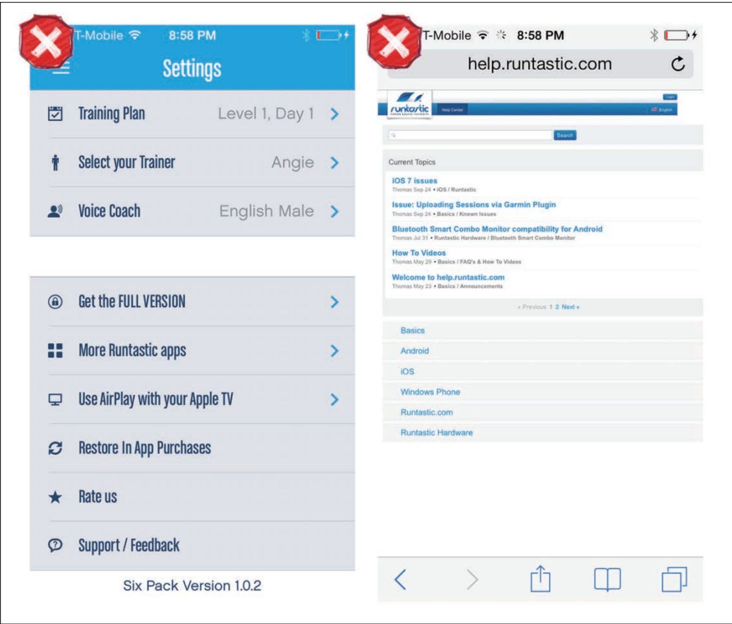



图 10-5
iOS 版 Runtastic's Six Pack 应用打开了桌面版网页，而不是移动优化过的网页版本

## 入门介绍

入门介绍是简单精炼的说明，解释这个应用是怎样运作的。入门介绍可以是一个单页，比如 Gyft 和 Cash 应用；也可以是一个视频演示，比如 Bump；还可以是更复杂的帮助系统的一部分，就像 Pocket 中设计的那样。

［小贴士］

入门介绍模式非常适合解释一步一步的工作流。入门介绍的描述要言简意赅。

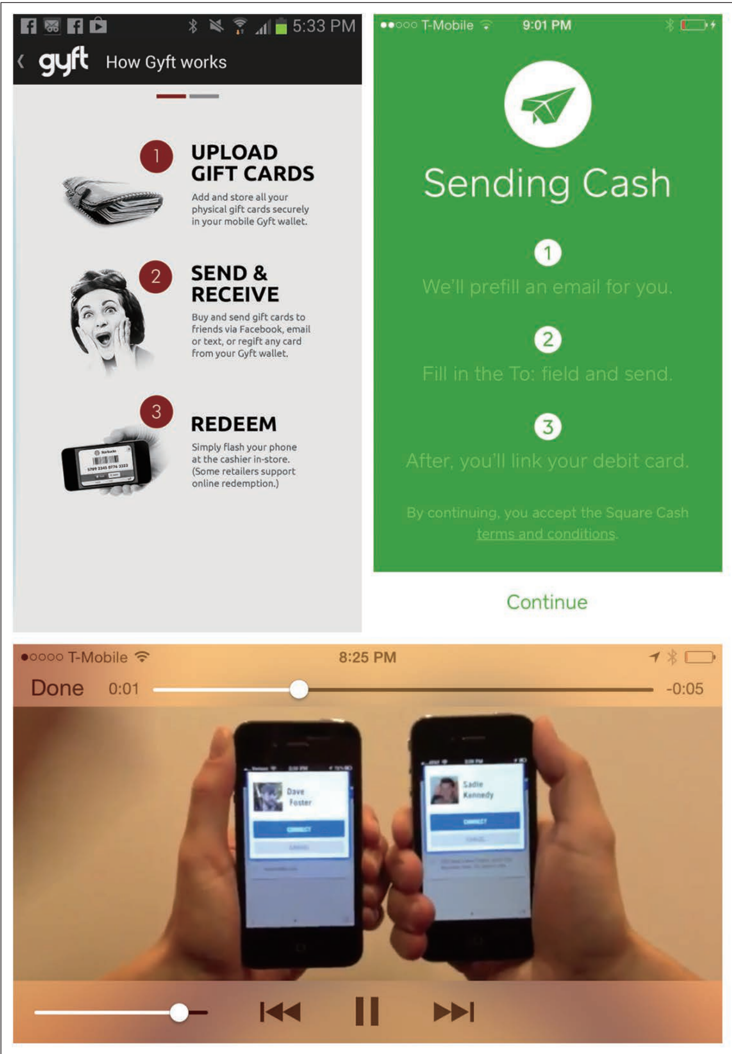
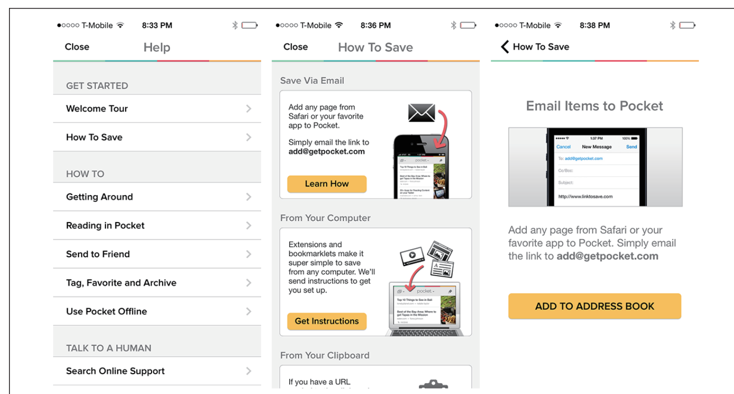



图 10-6

安卓版 Gyft 和 iOS 版 Cash：简单的入门介绍页面；iOS 版 Bump：入门介绍视频

图 10-7
iOS 版 Pocket：入门介绍结合了截屏、插图、文字和视频

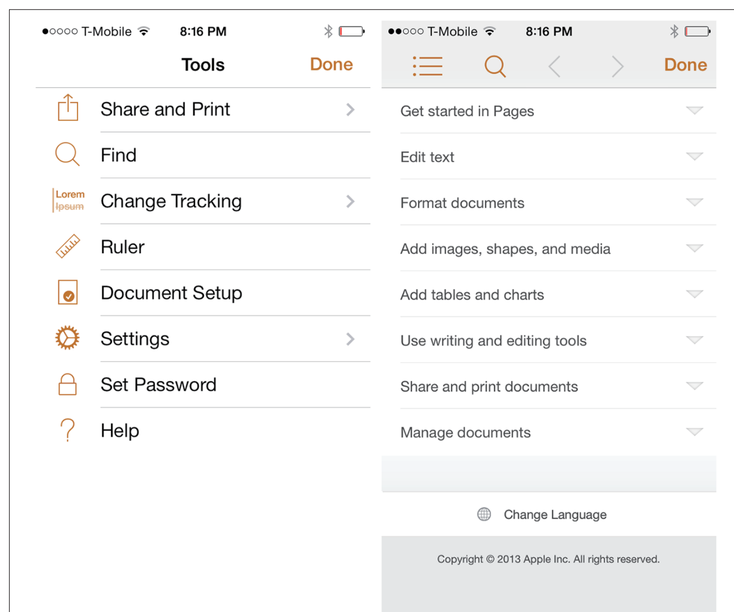



我在第 7 章里提供了一些提示，告诉你怎样做能使你的应用更能吸引用户。

## 用户指南/帮助系统

用户指南一般比简单的入门介绍层次更深。Pages 和 Facebook 在应用中内置了大量的多层级用户指南。

图 10-8
iOS 版 Pages：多层级用户指南



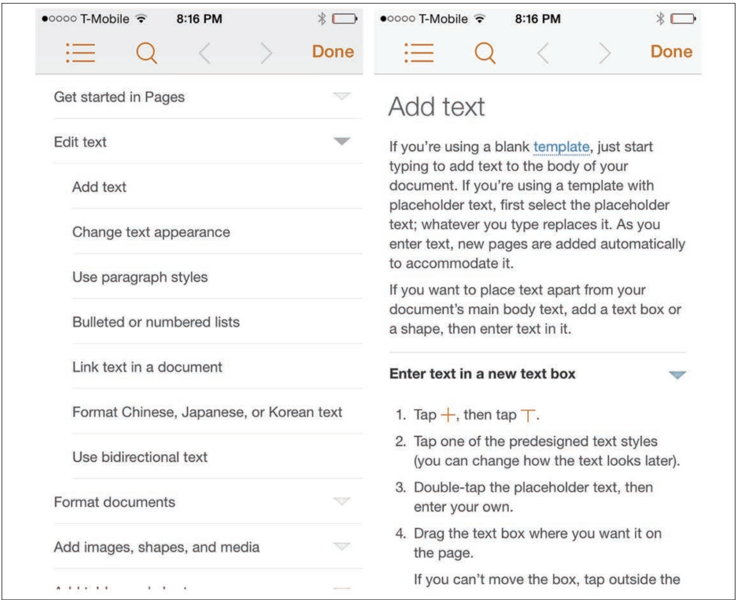



图 10-8（续）

iOS 版 Pages：多层级用户指南

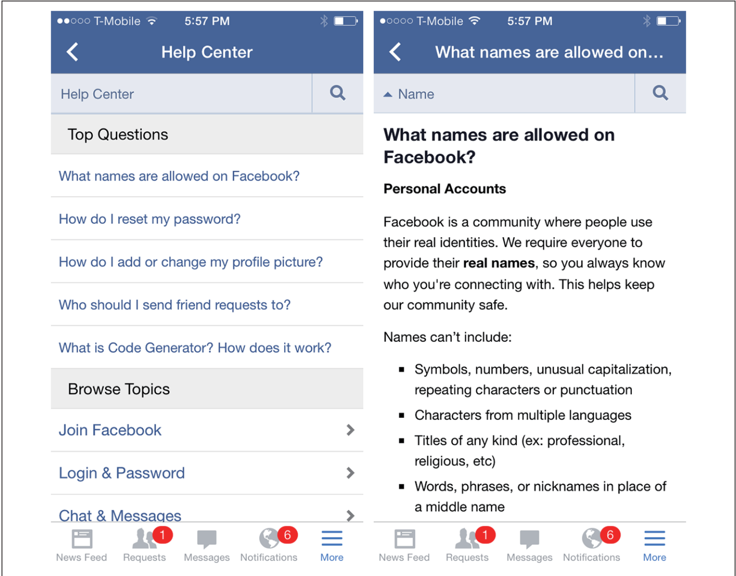



图 10-9

iOS 版 Facebook：多层级用户指南

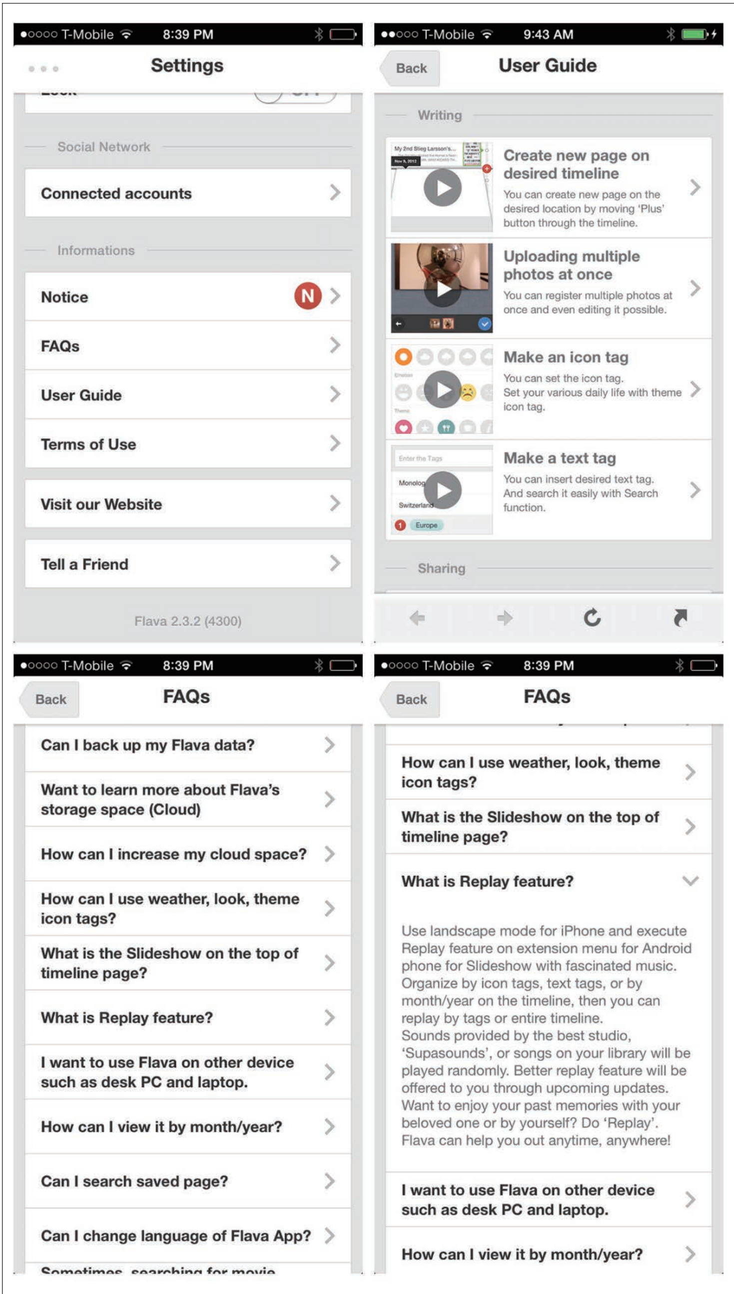
与 Facebook 类似，Flava 应用也提供了组织有序且易于导航的用户指南和常见问题解答。

［小贴士］

按照逻辑类别或主题编写用户指南，并提供内容搜索功能。

图 10-10

iOS 版 Flava 包括了用户指南和常见问题解答



## 常见问题解答

如果你想告诉用户的信息用一个入门介绍不足以表达，又不够一部完整的用户指南，怎么办？编写一个常见问题解答列表可能就能解决问题。常见问题解答通常把最常见的问题排在前面。（顺便提一下，这些问题可以通过对应用的预览版进行用户测试来确定。）Dropbox 使用了标准的 iOS 模式，逐级深入到每个常见问题解答中去。

［小贴士］

常见问题解答列表应当设计得易于浏览，应确保可简明直观地显示答案。把最常见的问题排列在前面。

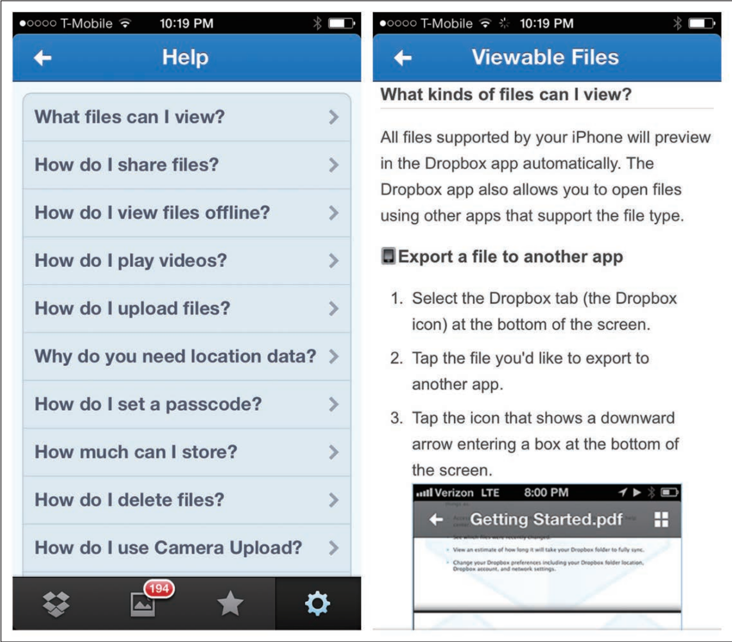


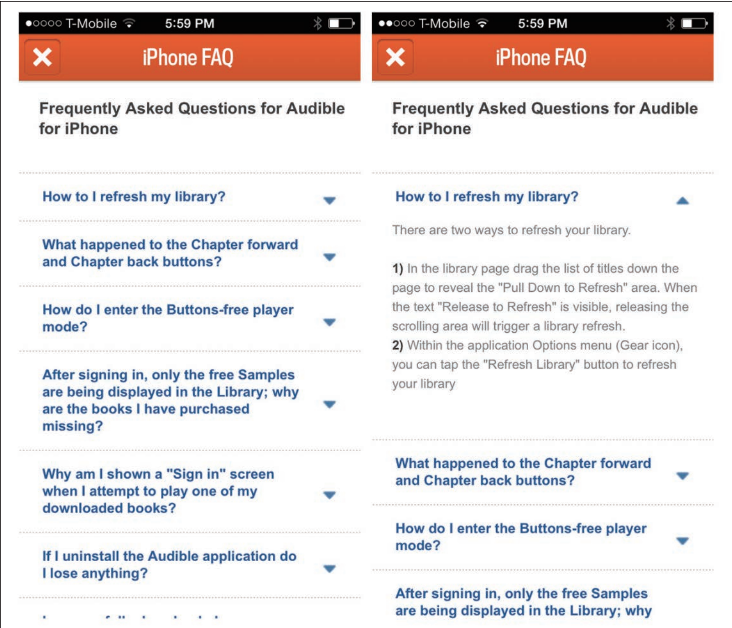

图 10-11

iOS 版 Dropbox：逐级深入的常见问题解答

Audible 在显示常见问题解答的答案时使用了行内展开的模式代替逐级深入。

图 10-12

iOS 版 Audible：常见问题解答在页面上展开

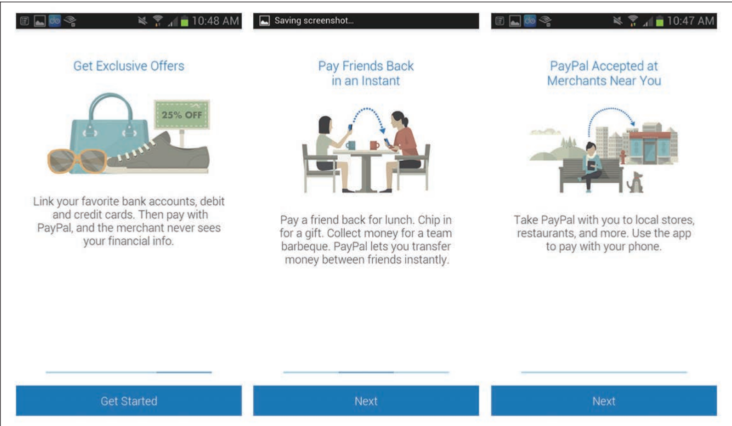



# 功能使用向导

功能向导可以用来突出一个应用的某些特性和功能。不要把功能向导和产品向导混淆，产品向导突出的是应用本身的价值点，比如下面的 PayPal 示例。

图 10-13

安卓版 PayPal：产品向导传达应用本身的价值点



功能向导通常在初次使用时显示，但之后也可以访问。在 Fidelity 中，可以通过“回看说明指南”（Bring Back Instructional Guides）设置开启向导。

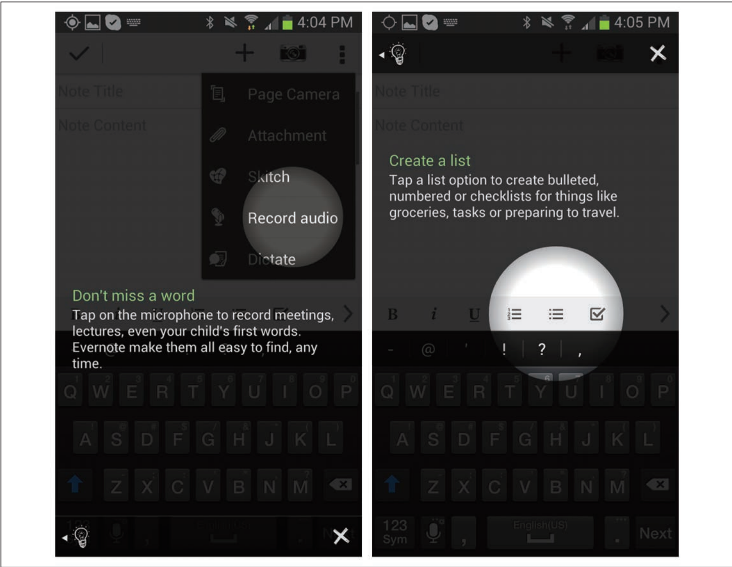



图 10-14
安卓版 Evernote：功能向导突出特定的功能和 UI 控件

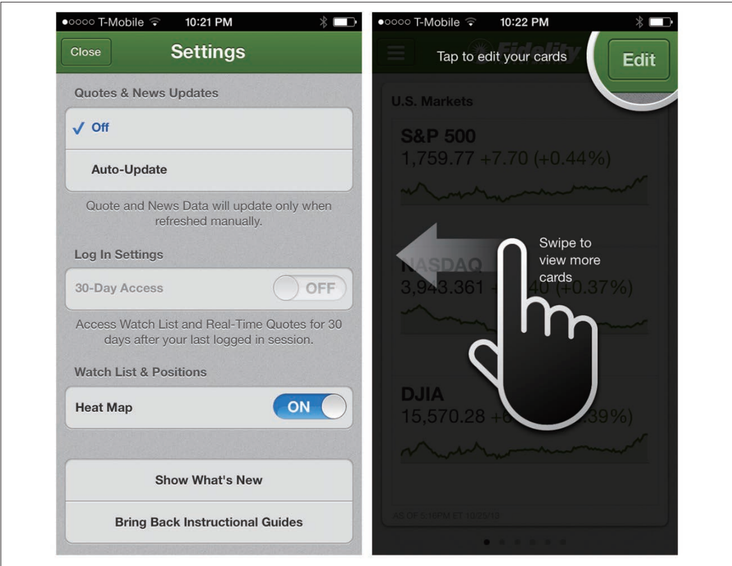



图 10-15
iOS 版 Fidelity：重启向导选项——“回看说明指南”

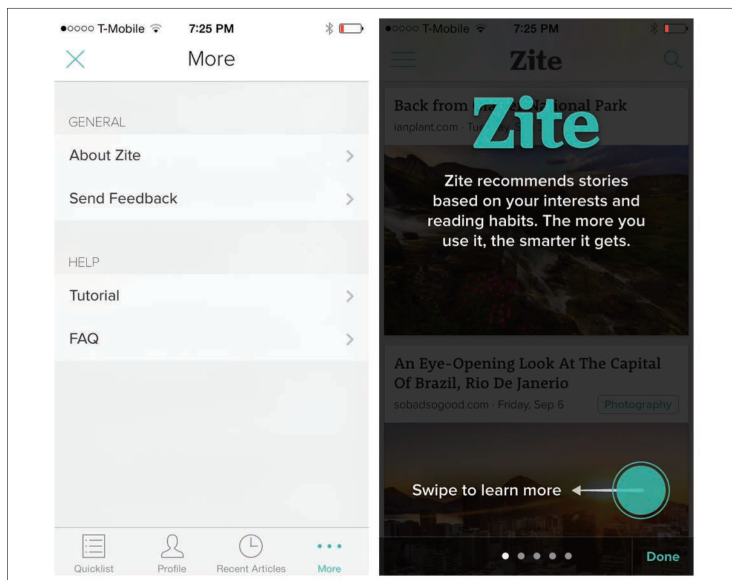



图 10-16
iOS 版 Zite：可以点击“更多”菜单中的选项重启向导

[小贴士]

推出功能向导可能会对应用的转化率和使用量产生负面影响。可以查看第 7 章中讲到的交互式引导设计的最佳实践。

虽然功能向导看上去挺漂亮，但用户测试表明，即便这些向导设计得很好，用户也会经常跳过去不看。当用户想直接进入应用时，它们还挺碍事，令人恼火。

最好的帮助设计依应用和受众的不同而不同，所以一定要测试多种帮助模式。可以回顾第 7 章中讲到的准则来避免无效的引导设计。

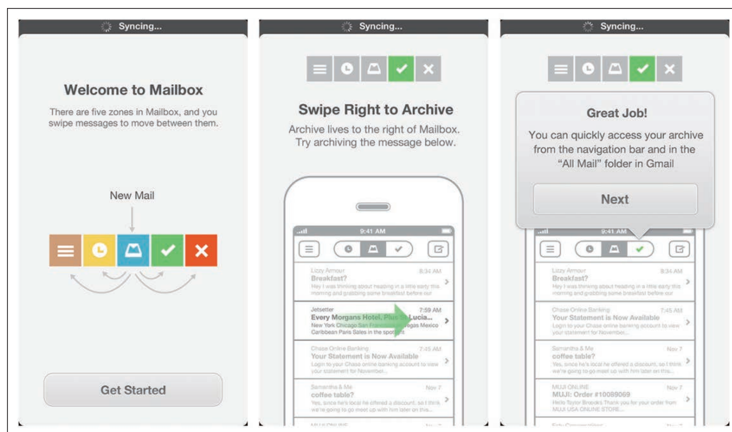



图 10-17
iOS 版 Mailbox 没有使用功能向导，而是设计了一个交互式引导（http://www.youtube.com/watch?v=URd0j5vEsHE）




图 10-18
作为对比，安卓版 Boomerang 中的功能向导真是铺天盖地、势不可挡

## 新手引导

一个精心设计的引导会让用户感到舒服并愿意对产品倾注心思，这一点我们在第 7 章中讨论过。虽然功能向导和新手引导在某些方面类似，但二者的区别在于，新手引导其实是使用户边使用应用边学习。这种模式非常适合需要初始配置的应用，或者依赖多步骤互动程序的应用，比如 Pocket 和 Vine。

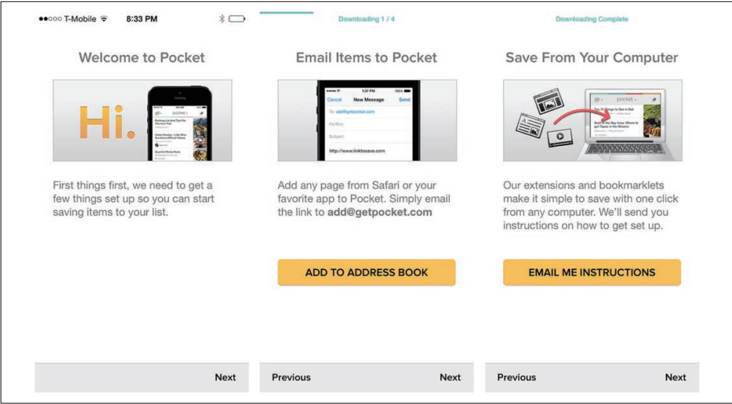
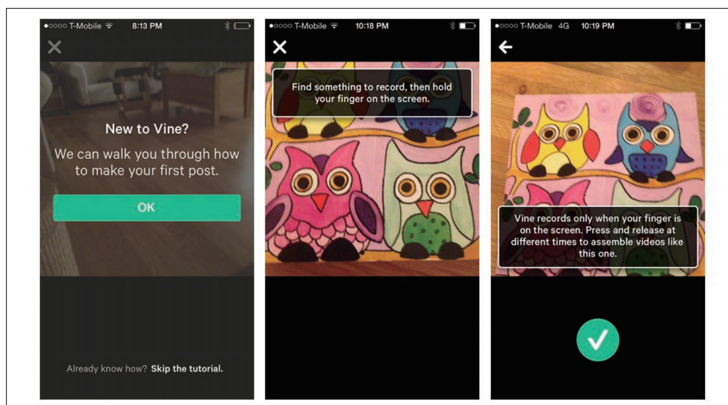


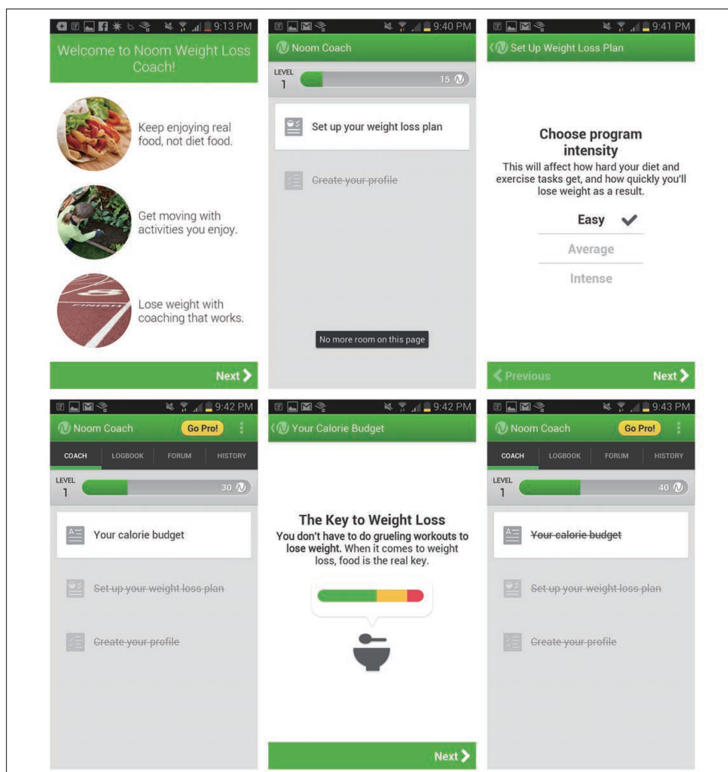

图 10-19
iOS 版 Pocket：新手引导

图 10-20
iOS 版 Vine：一般模式的新手引导（http://www.youtube.com/watch?v=vBOC9DpC5ns）




在 Noom 和 Everest 这类基于目标的或行为修正的应用中，新手引导既用作着陆页也用来留住用户。

图 10-21
安卓版 Noom：一个值得注意的新手引导



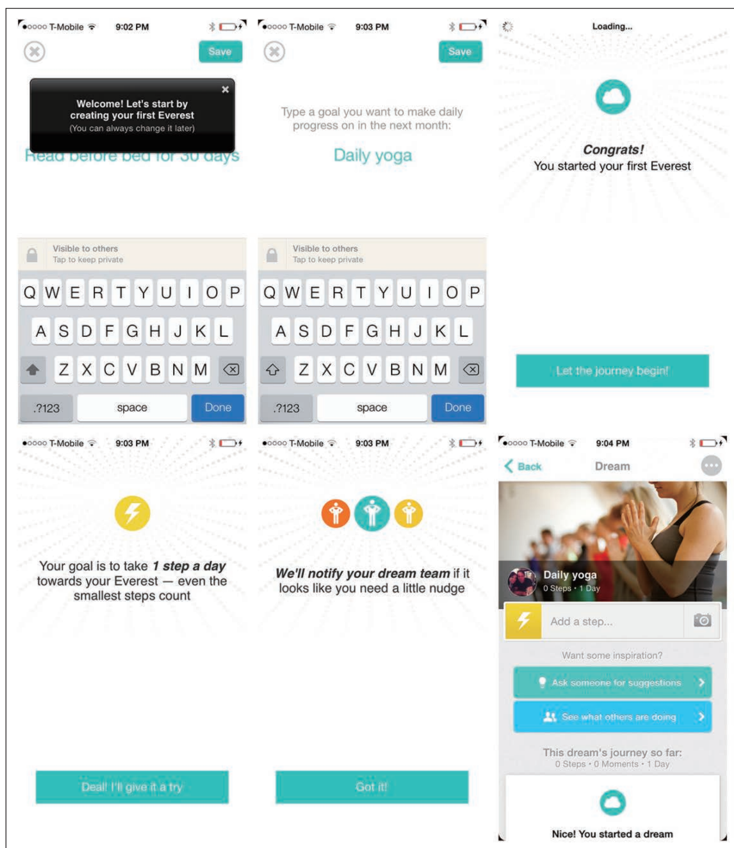



图 10-22
iOS 版 Everest：带有互动提示的新手引导

# 上下文帮助

上下文帮助是指在应用的特定位置获得的帮助。不同于功能向导的是，上下文帮助没有前置引导信息；这种帮助可在页面内访问，或在执行一项任务的过程中显示。

在 Argus 中，点击水滴会显示增加水量或减少水量的提示。

图 10-23
Windows Phone 版谷歌应用：用户可以点击特定的按钮，显示上下文帮助

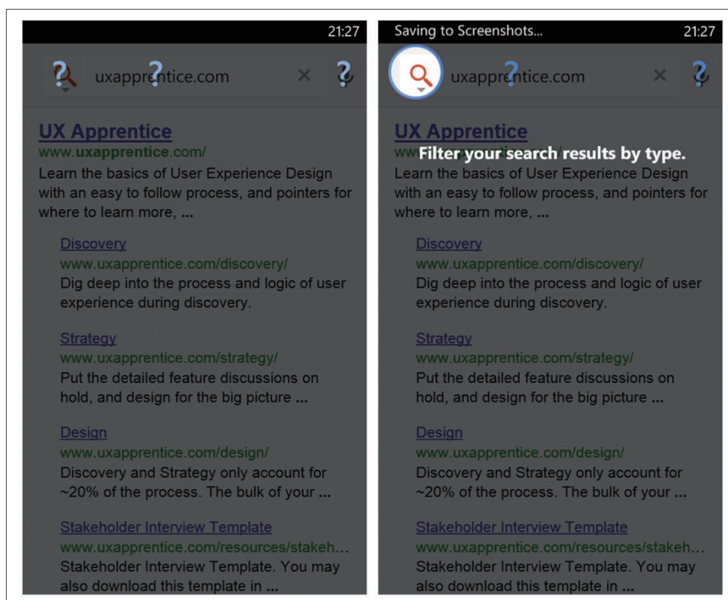



图 10-24
iOS 版 Argus：点击水滴会显示上下文帮助（http://youtu.be/FxvqBhs-2Co）



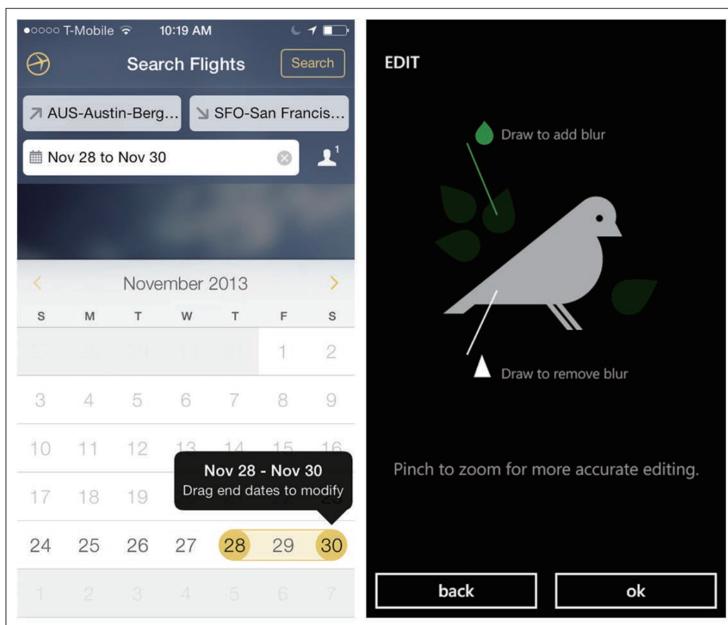



图 10-25
iOS 版 Expedia 和 Windows Phone 版 Creative Studio 在任务流程中提供帮助

## 收集反馈

程序 bug、错误、反馈、建议、增加新功能的需求——幸运的话，你的用户会愿意与你分享这一切。一定要为用户提供一个简单的机制来收集他们的反馈。

大多数应用在“设置”下都会有一个“反馈”选项，点击该选项会打开移动设备的邮件客户端，或基于网页的邮件表单。理想情况下，网页表单应该嵌入到应用内，而不是启动移动浏览器。

图 10-26
iOS 版谷歌应用：嵌入式收集反馈表单

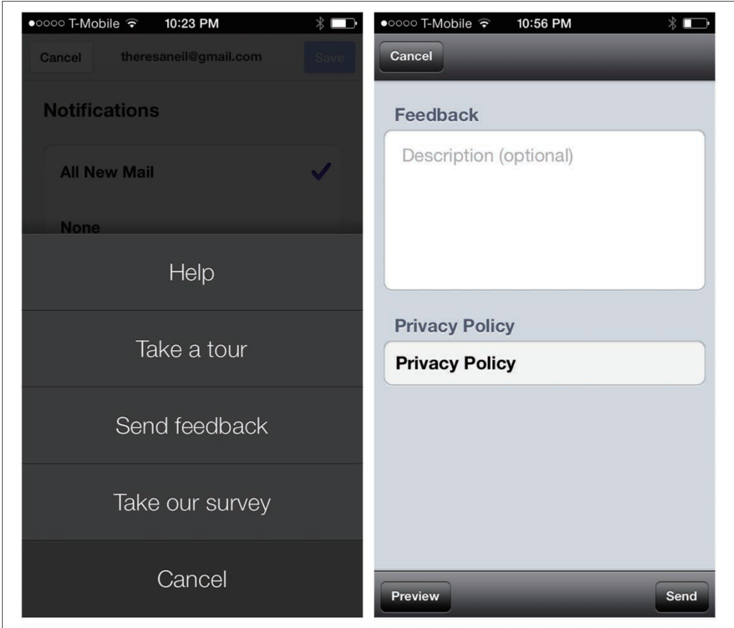
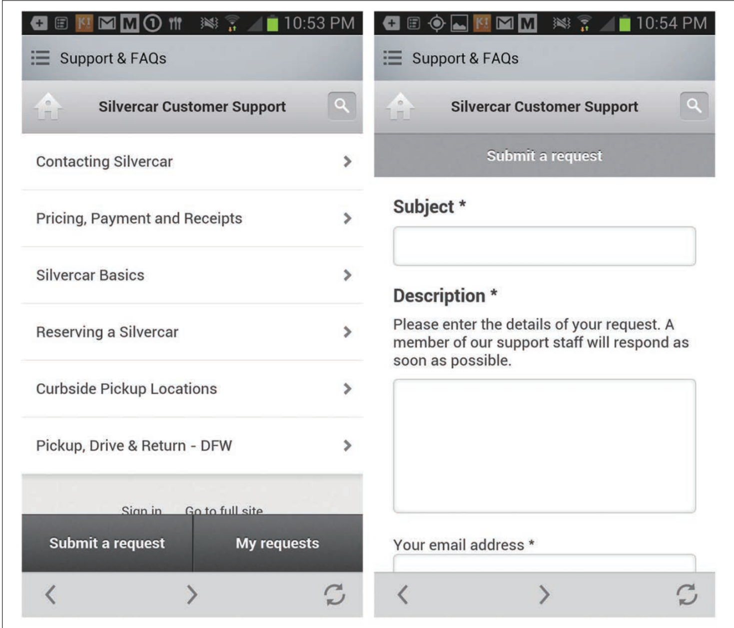



图 10-27
安卓版 Silvercar：嵌入式收集反馈表单（由 Zendesk 提供）



Wunderlist 整合了流行的网页端工具 UserVoice，创建了一个互动社区，用户可以在里面提建议，还可以对新功能需求进行投票。Get Satisfaction 是另一个基于网页的用户反馈工具，可以集成到原生应用中。

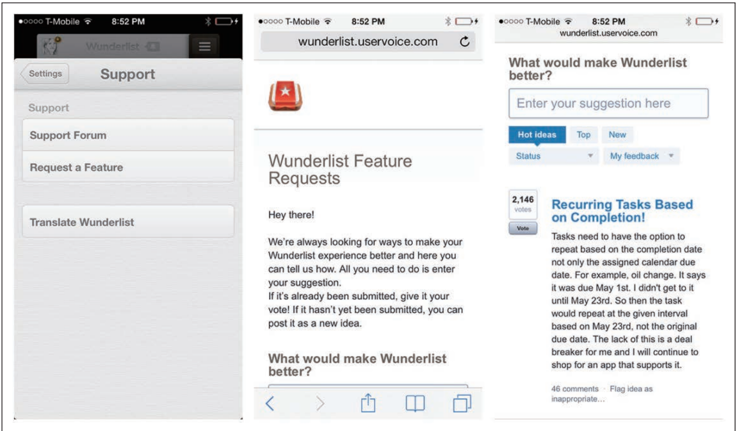



图 10-28

iOS 版 Wunderlist：用集成了 UserVoice 的收集反馈方式创建社区

**[小贴士]**

如果你要求用户提供反馈，就要做好接受的准备。提前做好计划，以便对用户的问题和功能需求做出优先排序和回应。

第 11 章

# 反模式



- 反模式

  标新立异（Novel Notions）、不必要的复杂（Needless Complexity）、隐喻错位（Metaphor Mismatch）、愚蠢的对话框（Idiot Boxes）、图表垃圾（Chart Junk）、按钮海（Oceans of Buttons）、格格不入（Square Peg，Round Hole）

什么是反模式？维基百科的定义为：

> 反模式也称为陷阱（pitfall），指一类通常是重复发明的糟糕的问题解决方案。反模式已被当作一个研究类别，以便日后加以防范，并在调查系统故障时加以辨别。这一术语源自计算机科学，很明显，很明显其灵感来自四人组（Gang of Four）[1]的《设计模式》（*Design Patterns*）一书，此书介绍了一些高质量的编程方法。

与计算机科学类似，以下这些设计上的反模式也是应该避免的常见设计陷阱。由于每个重新发明都有其各自创新方面的毛病，我不能一概而论，所以现在就让我们直奔主题吧。

## 标新立异

新手设计师总想尝试前卫的、原创性和革新性的设计。但大部分情况是，设计出来的效果很差，让人难以理解，更难上手使用。有时候，造成反模式的原因是懒惰或坏习惯。按照可用性挑刺专业户 Richard Gunther 的说法就是："你通常能够判断一个移动应用开发团队是否具有传统网络开发背景。这些团队常常会尝试把旧有的交互模型转换到新平台上，并且设计出一些非标准的 UI 元素，号称那是他们的'创新'。"

标新立异在移动应用中随处可见，大到主导航，小到一个单独的控件、交互或手势。NBC News 以其孔雀开屏般的转盘菜单在我的标新立异式榜单中位列榜首。谢天谢地，大多数应用都没有试图创造一个与自己的商标形状一样的菜单。




图 11-1
iOS 版 NBC News：以公司商标为灵感的导航

注 1：指《设计模式》一书的四位作者 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides。（译者注）

那么为何它就是反模式呢？让我们逐一道来：

- 不管是对普遍的移动平台来说，还是对于 iOS 平台，它的菜单位置都不标准。比如，作为下拉菜单的导航图标应该放在导航栏的一角，而不应该放在中间。
- 旋转的文字让人难以阅读。一个简单的左对齐分类列表读起来会更省时间。
- 颜色太多了！过度填充的色板让人迷惑，同时也降低了菜单选项的易读性。
- 这个设计在导航的基础功能方面就失败了，导航的基本功能就是要让用户看到自己当前在哪儿以及能去哪里。一旦关闭这个菜单，无论是选中的话题标题还是节目标题都不再显示在页面上。

USA Today 的设计就没有 NBC News 那么花哨，却易用得多。USA Today 设计了一个彩色且实用的侧边抽屉模式导航系统。每个类别的新闻都用不同颜色表示，但不是一次都显示出来，这样就避免了孔雀开屏似的效果。

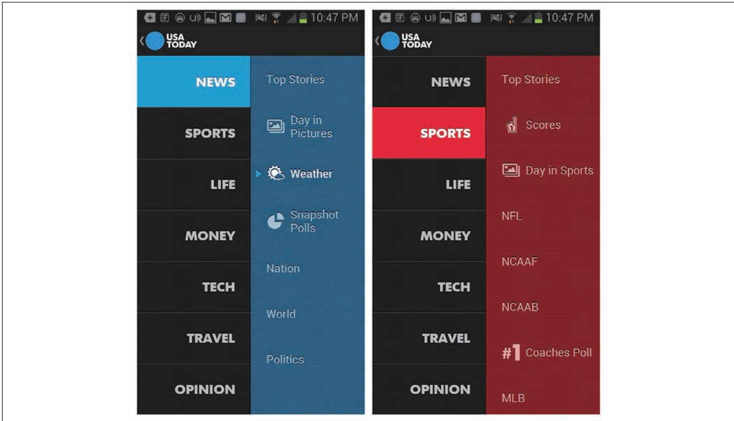



图 11-2
安卓版 USA Today：菜单中明智的色彩运用

阿拉斯加航空应用的菜单设计也令人难以捉摸。它设计了一个可以从页面顶部下拉的抽屉。我最后才意识到，这应该是模仿飞机上窗户的遮光帘！这是一个过度拟物设计（即对现实事物的隐喻）的很好的例子。这样的设计对用户有帮助吗？没有，它仅仅是为了讨人喜欢才这么设计的。

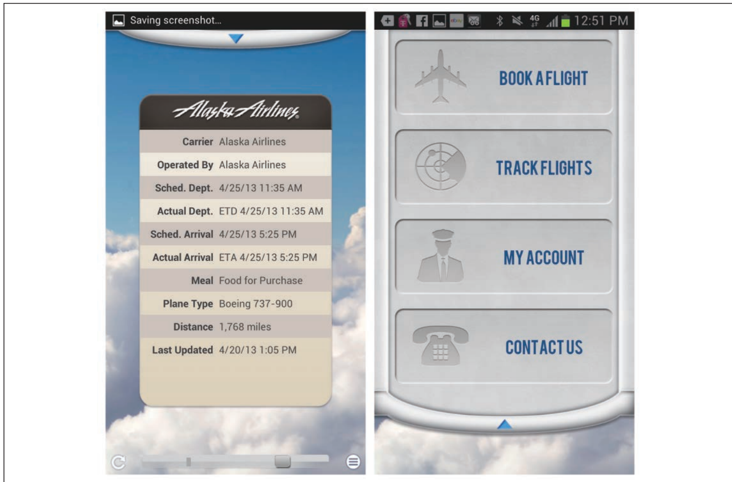



图 11-3

安卓版阿拉斯加航空：遮光帘式菜单设计对用户没有帮助

然而，遮光帘式菜单并不是这个应用唯一的标新立异之处。机票搜索结果以浮动的卡片形式显示出来。通过卡片下方的滚动条可以切换卡片——如果你看得见的话。拖拽手柄会显示每一项的更多信息，但手柄的位置看上去和卡片没有任何关系。




图 11-4

安卓版阿拉斯加航空：另类的搜索结果视图；用底部的滚动条切换卡片

想要获得更简单、更快捷的订票体验，就试试 Expedia 吧。Expedia 拥有标准的导航、简洁的搜索表单，以及创新（且易用）的查询和选择机票设计。

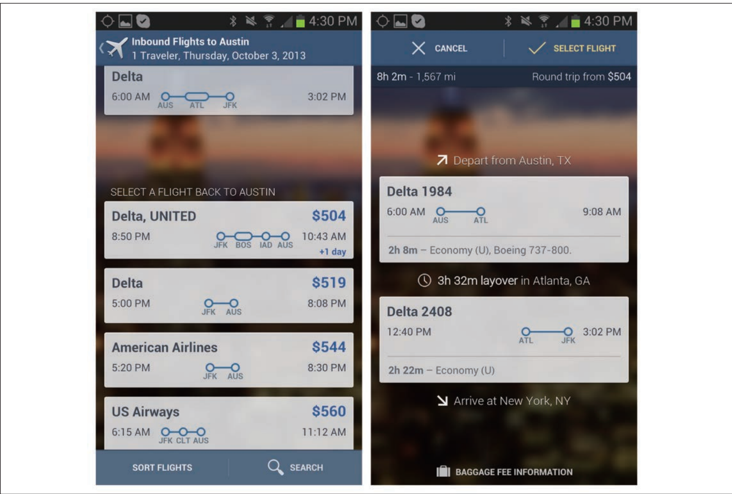



图 11-5

安卓版 Expedia：优秀的航班航段和航班中转视觉化设计

Hipmunk 也是一个订购机票的创新设计示例。注意，创新是针对用户体验层而言（把“agony”作为一个搜索排序选项）[1]，而不是在 UI 控件层面（一个窗帘式菜单丝毫不能改善我的订票或旅行体验）。

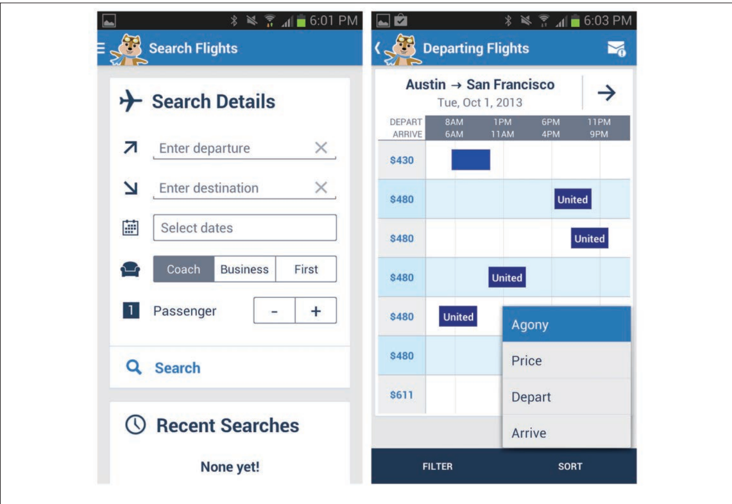



图 11-6

安卓版 Hipmunk：熟悉的控件带来创新的体验（按照 agony 排序）

注 1：Agony 是 Hipmunk 的一个智能排序，应用中官方解释为：“Agony 是我们独有的机票排序，它能帮你找到票价最低、飞行时长最短且中转航班最少的机票。”（译者注）

还有一个来自 Twitter @music 的反模式例子。设计师们把几种导航模式结合起来，设计了一种“创新”界面。在这句话中，我讽刺性地用到“创新”一词，意思是他们花了更多的时间和金钱却设计出一种降低了可用性的自定义控件。

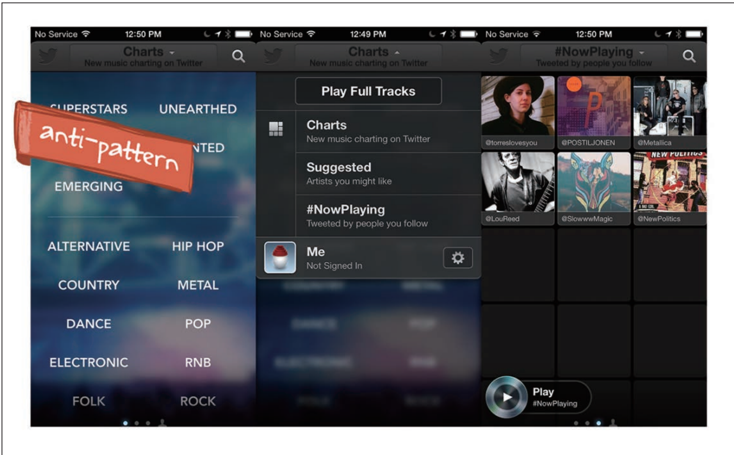



图 11-7

iOS 版 Twitter @music：安卓风格的下拉菜单 + 页面指示器 = 过分之举

导航栏设计有一个自定义下拉菜单，页脚有页面指示器（4 个小圆点）。让人惊讶的是，下拉菜单和页面指示器是相互关联的，选择下拉菜单中的一个选项将会翻到下一页。但等一等，还不只如此！向左或向右滑动还会更新下拉菜单，显示出最新选中的模块。

你是不是觉得他们已经知道这个“下拉菜单加页面指示器”控件是个不好的设计？如果是的话，至少可以解释为什么他们把导航栏设计成超低对比度、黑色叠加黑色的样子——也许他们就是在努力掩盖那个控件啊！

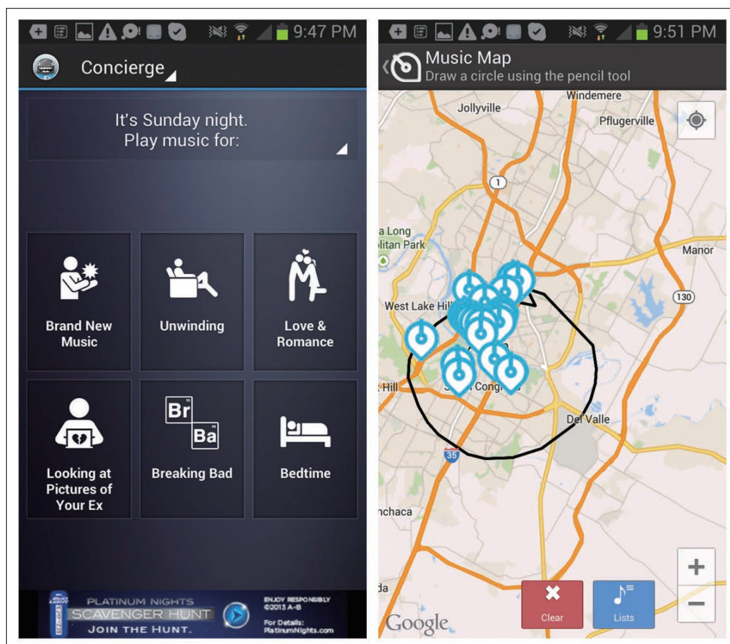
Songza 和 Soundwave 是音乐类应用中真正有创新设计的例子。Songza 的礼宾部（concierge）隐喻实在绝妙[1]，而 Soundwave 让你可以探索结合社交元素和地理位置元素的音乐地图。

[小贴士]

如果你想在你的应用中做出创新，那么就要在改进整体用户体验的概念模型和流程上下功夫，寻找帮助用户的机会。在设计 UI 之前，先要理解系统设计指南，熟悉已有的设计模式。

注 1：Songza 的 concierge 是根据不同的时段、活动等，提供个性化场景音乐的功能。（译者注）

图 11-8
安卓版 Songza 和 Soundwave：没有反模式的创新




## 不必要的复杂

很好的用户体验和很差的用户体验之间的区别之一在于效率。用户体验指导 Whitney Hess 说："效率涉及提高生产力和减少所需精力两个方面，一个流畅的设计能让用户在同样的单位时间内完成更多事情。创造效率表明你非常尊重客户，而他们也一定会注意到你的努力。"

让我们来看看一些有着不必要的复杂界面的应用，这些应用限制了用户的使用效率。

Eureka Offers 是一个寻找当地和全国优惠活动的简单应用。用户可以保存他们喜欢的优惠券。Google Offers、Shopular、SnipSnap、RetailMeNot 以及其他大多数这类应用全部用简单的行内操作实现保存、存储纸质优惠券和收藏优惠券，而 Eureka Offers 与它们不同，用户需要通过长按然后拖放的交互才能实现保存操作。这种设计不但让用户更费力，而且也比行内操作更隐蔽（见第 5 章）。

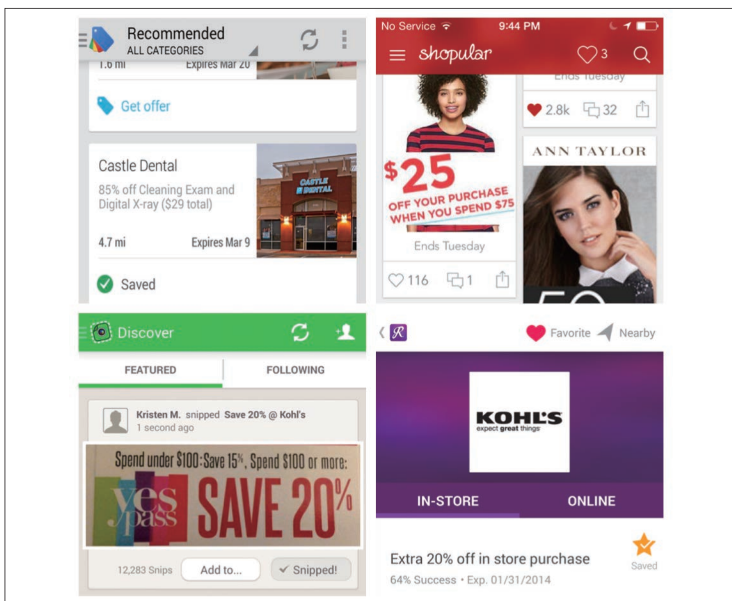



图 11-9

Google Offers、Shopular、SnipSnap 和 RetailMeNot 全部采用了行内操作实现保存、存储纸质优惠券和收藏优惠券

另一个采用拖放交互、完全不考虑操作简便与否的例子是 Facebook 的聊天头像。为何不设计成“长按查看 / 隐藏”选项，或者滑动出屏幕来删除，又或者至少不在页面上显示浮动的头像？

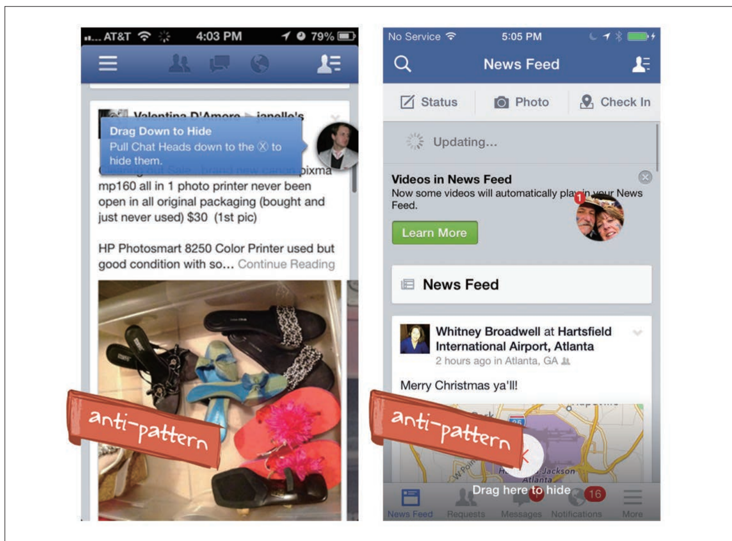
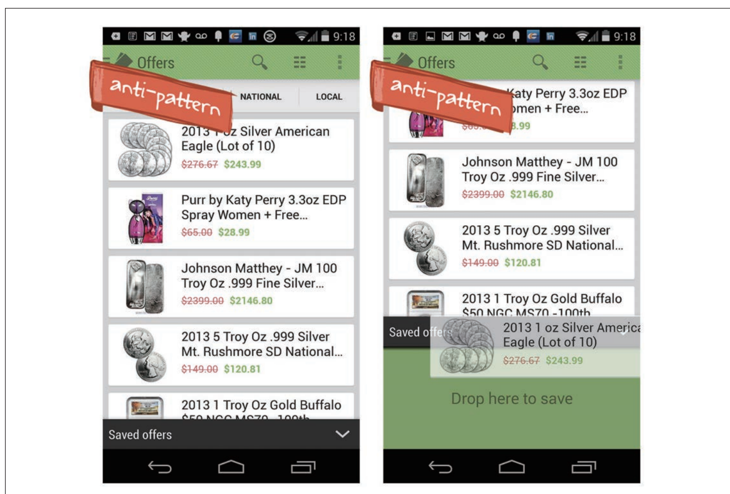



图 11-10

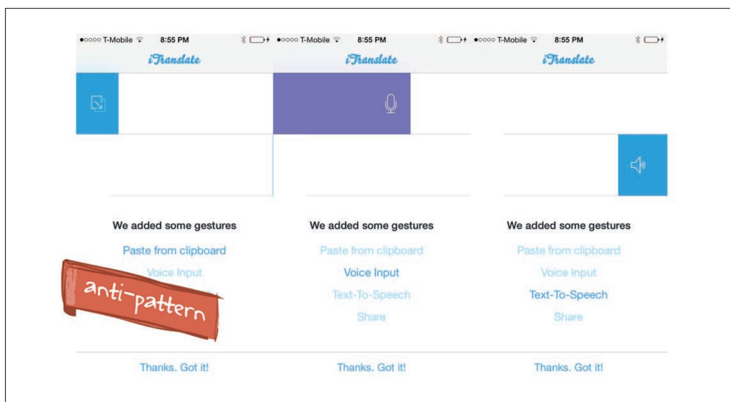
iOS 版 Facebook：聊天头像并不直观，而且从视图中删除头像更麻烦

图 11-11
安卓版 Eureka Offers 需要长按并拖放才能保存




iTranslate 应用的设计师显然爱上了基于手势的创新 UI 设计。我这么说是因为这个应用到处都需要你使用非直观的手势。向右短距离滑动以从剪贴簿粘贴，向右长距离滑动以使用语音输入，搞什么呀？

图 11-12
iOS 版 iTranslate：UI 设计充斥着非直观的手势




设计师可以提供简单的方式让用户选择输入模式，就像 Amazon Price Check 应用中的“扫描、拍照、语音”（Scan It，Snap It，Say It）那样。实际上，iTranslate 的一个竞争对手，Google Translate，就在文字输入框下面提供了“拍照、语音、手写”（Snap It，Say It，Write It）的上下文工具栏。

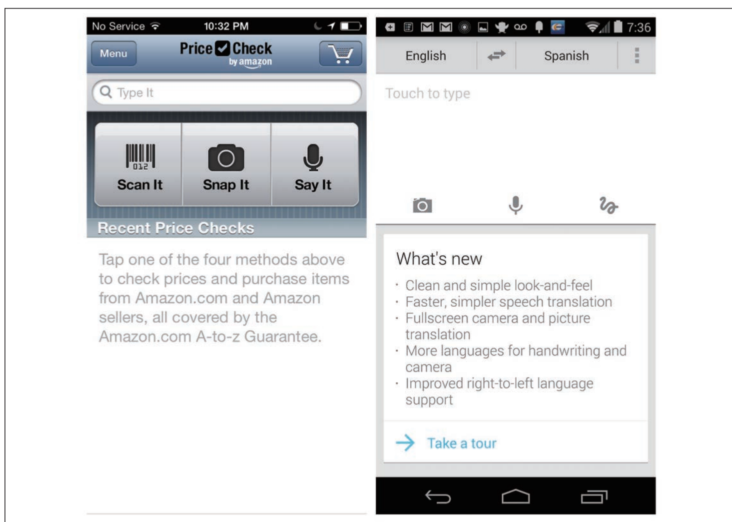



图 11-13
iOS 版 Amazon Price Check 和安卓版 Google Translate：直观的输入模式选择器

在接下来的例子中，我们可以看到，Over 应用的设计师实在是太喜欢转盘式菜单了。菜单可以在首页上从右向左滑出。这个设计理念倒是没什么错，但不应以它这种方式实现。转盘式菜单选项把主导航和操作结合在了一起，而且其选项总在变化。这是我所尝试过的真正令人不安的照片编辑工具，可能也是最没有效率的一个。

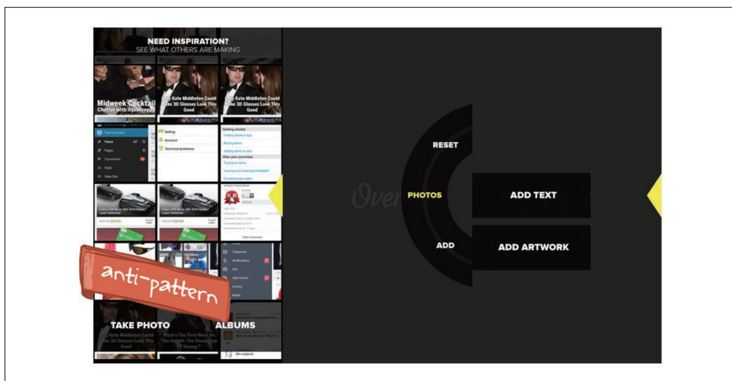



图 11-14
iOS 版 Over：令人不愉快的导航创新（http://www.youtube.com/watch?v=rMG0ftTdSy4&feature=youtu.be）

你应该时常问问自己："我能不能进一步简化设计？"如果答案是"可以"，那么就照做吧，即使（或者可能尤其）这样意味着牺牲创意。但你的用户会对此表示很喜欢。

## 隐喻错位

造成这种反模式的原因是在界面设计上选择了错误的隐喻。隐喻错位可能出现在低层次，比如没有正确使用一个控件或图标，也可能出现在高层次，比如应用的概念模型与用户的心理模型不符。

## 控件错位

Conqu 和 DailyBurn 就是低层次控件错误的示例。出于未知的原因，这两个应用的设计师都选择创建自定义控件，而不是使用安卓系统的默认控件。Conqu 使用了一个 iPad 弹出控件，而 DailyBurn 则使用了 iOS 风格的选择器。

图 11-15

安卓版 Conqu 和 DailyBurn：在安卓应用中使用了出人意料的 iOS 风格控件




iOS 和安卓版 Edmunds 的主导航采用了非标准控件。如果用户不能轻易看出应用的主模块，那么这就会影响应用的使用率。

Edmunds 应用的首页默认几乎是空白的。我过了好一会儿才意识到那是一个搜索表单。这个应用的其他模块也很容易被忽略，因为在安卓版应用中它们被塞在了下拉菜单控件下面，在 iOS 版中则塞在了一个箭头下面。

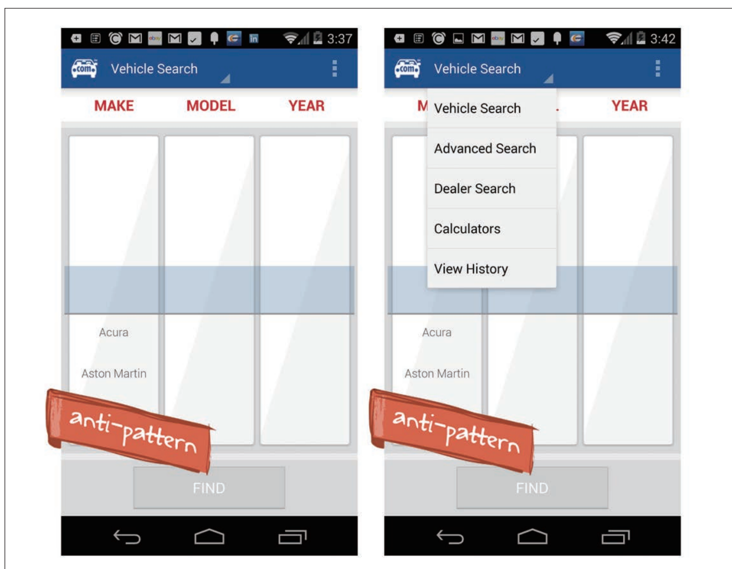



图 11-16
安卓版 Edmunds：首页，以及错误地使用下拉菜单作为主导航

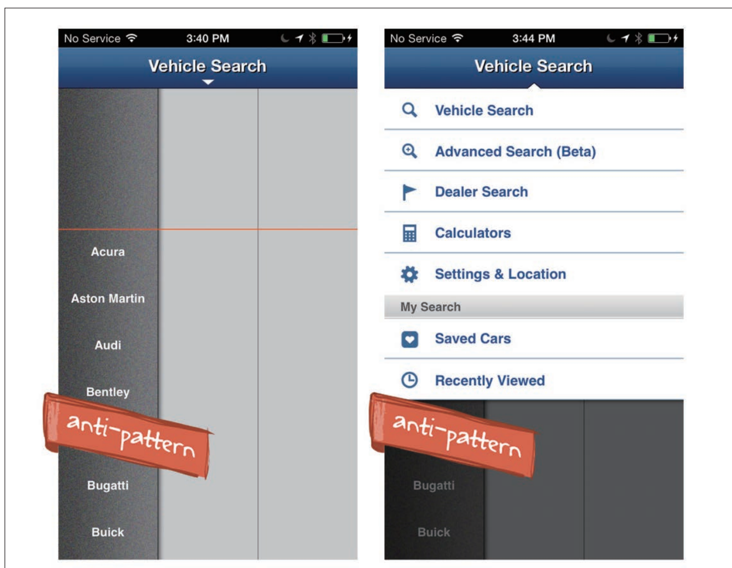



图 11-17
iOS 版 Edmunds：空白的首页确实是一个使用非标准菜单的搜索表单

一个简单的修正方法是使用导航选项卡进行切换（见第 1 章）。隐式搜索模式也很适合首页的车型搜索，不管怎样，总比一个空白的首页强。（Zillow 和 Trulia 的隐式搜索例子参见第 4 章。）

## 图标错位

用户对通用图标有特定的认识：齿轮代表设置，星星代表保存或标明收藏，放大镜意味着搜索，等等。把熟悉的图标用于其常用的功能之外就会导致混乱。比如 iOS 版 Kindle 应用，我好不容易才发现调节亮度的控件放在了字体设置图标内，真让人诧异。

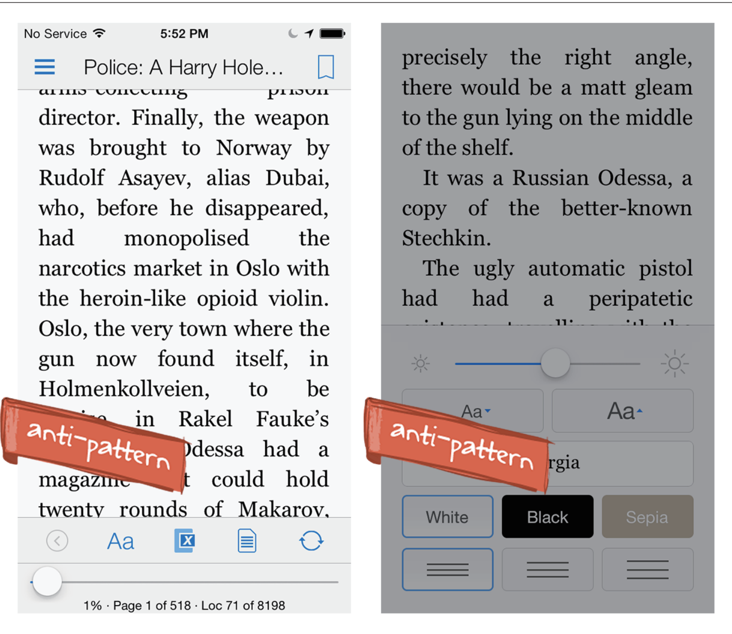



图 11-18

iOS 版 Kindle 把亮度控件隐藏在了字体设置图标下面

## 手势错位

“下拉刷新”现在已经是一个非常标准的手势了。我们能在生产力工具、社交和娱乐、购物等类型的应用中见到它。Tweetbot 3 和 Audible 是下拉刷新设计的典型示例。

邮件应用 Seed 试图教会我们一种新的招数：下拉添加提醒。可想而知，其结果事与愿违。我们都知道下拉是做什么用的，或者就当我们知道吧，总之下拉不是用来给邮件添加提醒的。我猜有人终于想明白了，因为现在最新版的 Seed 已经改用一个人们熟悉的闹钟图标表示提醒了。

［小贴士］

下拉刷新可能不久就会消失，因为现在大多数设备的配置都完全可以处理后台更新（信息来源：http://www.theverge.com/apps/2013/12/20/5231728/is-it-time-to-kill-pull-to-refresh）。

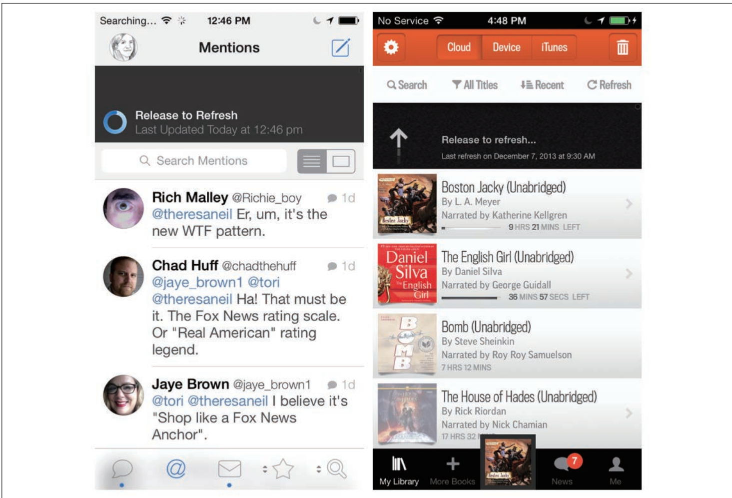



图 11-19

iOS 版 Tweetbot 3 和 Audible：下拉刷新已经成为标准手势

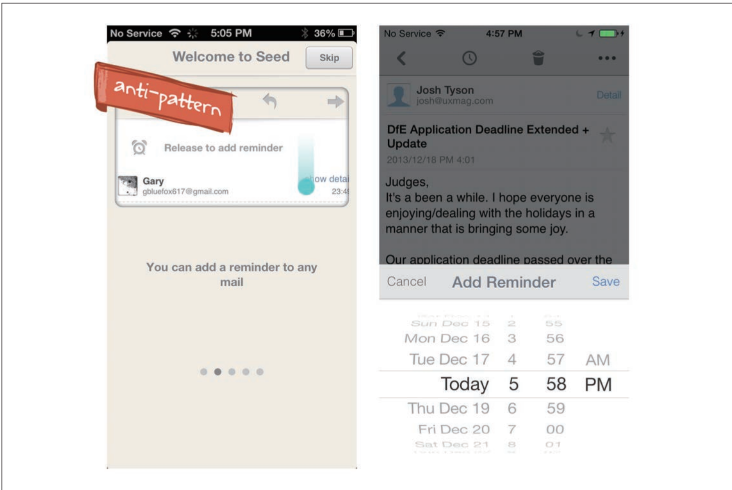



图 11-20

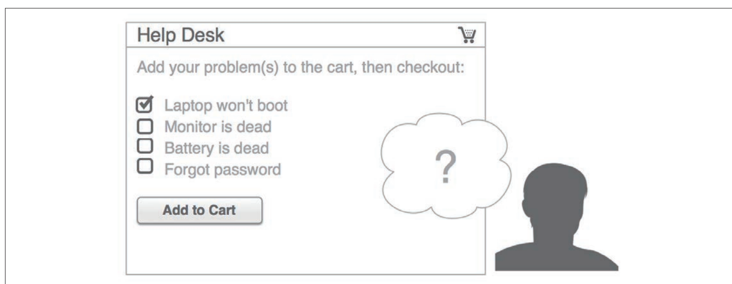
iOS 版 Seed 的前后两个版本：左图为下拉添加提醒，右图为改版后使用了提醒图标

## 心理模型错位

讨论这类反模式时，我最喜欢用的一个例子是我的前老板要求我们使用的一个故障申报系统。每当出现一个技术问题，无论是忘记密码还是笔记本死机，我们都要登录这个系统，浏览问题列表，然后选择一项问题，把它添加到购物车，然后结账。这是个多么好的电商应用设计啊，但是完全与技术支持应用不搭边！

图 11-21

心理模型错位：把申请技术支持当成了疯狂购物




[小贴士]

无论是图标、控件，还是概念模型方面的隐喻，都是可以简化和提升体验的视觉化快捷方式。但是使用不当的话，会让用户感到挫败且难以掌握。

另一个例子是 ABC News iPad 应用在 2011 年左右的版本。一个地球仪样子的新闻内容导航？我想，如果新闻报道是从地球仪上的特定地理位置显示出来，还有点道理，但实际上完全不是这样。ABC News 地球仪与我们赖以生存的地球毫无关系。它只是一个难以阅读、更难浏览的不停旋转的球。

对比来看看 Geo Walk 的地球仪。Geo Walk 是一个探索世界各地真相的教育类应用，它把真相卡片放置在地球仪上的不同地点，这样你就能探索某一个特定区域的真相。




图 11-22

iPad 版 ABC News（2011 年）使用地球仪作为新闻内容的导航；iOS 版 Geo Walk，显示地理信息的合理的地球仪设计

## 愚蠢的对话框

在《软件观念革命：交互设计精髓》（*About Face 3: The Essentials of Interaction Design*，2007 年 Wiley 出版社出版，http://amzn.to/1fFC7SK）一书中，作者 Alan Cooper 写道："在使用协调的工具进行工作时，一个人才能进入高效的心理状态。平白无故打断

用户的操作流是一种非常愚蠢的干扰行为。”

Bill Scott 也对“非常愚蠢的干扰行为”这一观点表示赞同，并且把这种行为命名为“愚蠢的对话框”反模式。

我认为我遇到过的“愚蠢的对话框”中最差劲的体验来自这样一种应用：第一次使用时就让你给应用评分，然后让你开启基于地理位置的服务，然后让你同意给你推送消息，然后又一次让你给应用评分。连续四次强制弹出对话框让我确实想给这个应用打分了——一个又大又圆的零蛋！

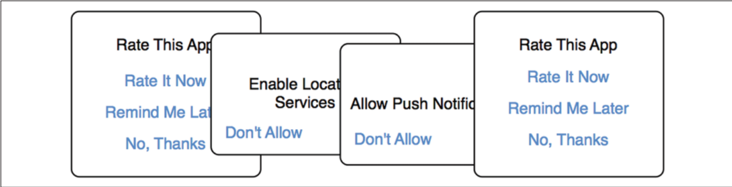



图 11-23
愚蠢的对话框：初次使用就弹出多个对话框，在正式开始前就扰乱了用户流

那些第一次使用就弹出对话框让你注册的应用更讨人厌。关于这种烦人的设计以及为什么这种设计没有必要，可以参看后面由 Greg Nudelman 撰写的“番外篇”。

Jamie Oliver's Recipes 应用也是一个愚蠢的对话框例子。在我点击“添加到购物清单”时，应用弹出一个对话框：“你想把所有这些食材添加到你的购物清单吗？”可不是么，要不然我为什么会点那个按钮呢？！

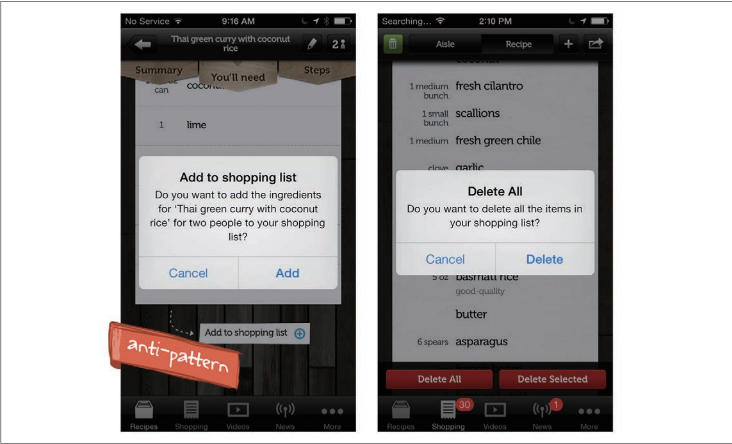


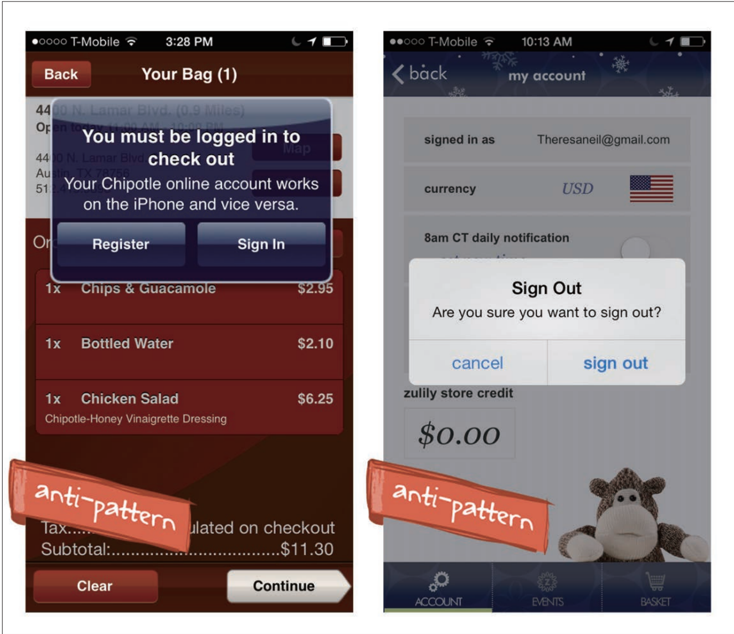

图 11-24
iOS 版 Jamie Oliver's Recipes：对每项操作都进行确认完全没有必要

第二张图示的“删除确认”对话框没有什么问题。在执行一个不可撤销的操作前要求用户确认是正确的设计，比如删除一个列表中的全部项，或是永久清空回收站。但没有必要对用户的每个操作都要求做出明确的确认。比如，对于添加到列表这个操作，使用瞬时反馈提醒用户该项已被添加就可以了。（参见第 9 章。）

Chipotle 是又一个愚蠢的对话框例子：在我准备结账的时候弹出了一个注册 / 登录对话框。为什么不直接转到登录页面呢？ Zulily 应用想象不到我有多想退出登录，所以确认了两次才允许我退出。傻死了！

图 11-25

iOS 版 Chipotle 和 Zulily：愚蠢的对话框淋漓尽致地发挥了作用，干扰了用户操作流




许多购物应用都使用愚蠢的对话框，给予商品已添加到购物车的反馈，比如 Haute 和 Rent the Runway。一种干扰较小的、更有效的方式是只增加标识购物车里商品数量的数字即可。你也可以显示一条行内确认信息，或是一个添加商品到购物车的动画。参见第 9 章的相应示例。

［小贴士］

不要扰乱用户的操作流。只在用户执行不可撤销的操作（比如永久删除），需要用户确认时才显示对话框。如果非要提示用户给你的应用评分，至少要等到用户有幸用上应用之后再说。

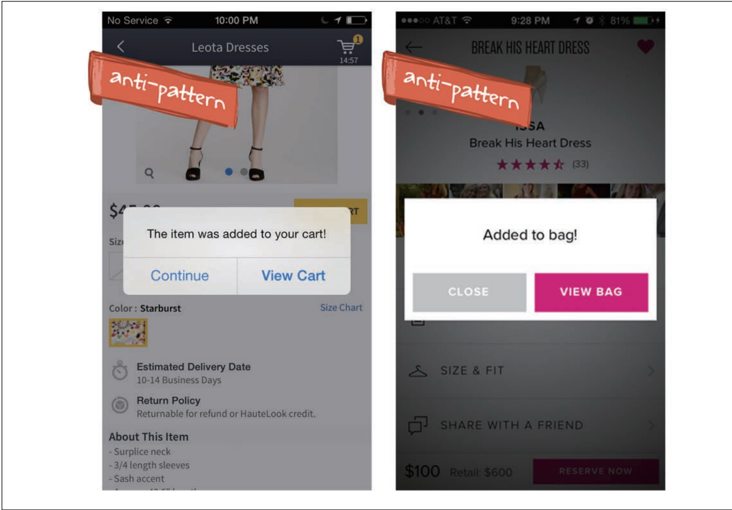



图 11-26
iOS 版 Haute 和 Rent the Runway：在添加商品到购物车时弹出愚蠢的对话框

# 图表垃圾

Edward Tufte 在他 1983 年的著作《定量信息的视觉展示》（*The Visual Display of Quantitative Information*，Graphics Press 出版）中，创造了“图表垃圾”一词，指的就是下图这样的图表。

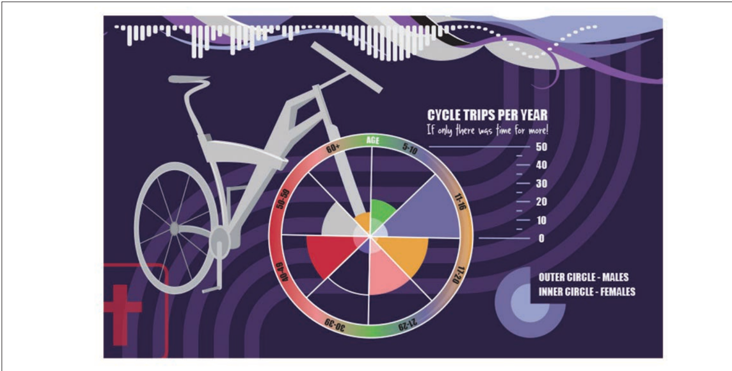



图 11-27
WTFViz 博客上的图表垃圾[1]

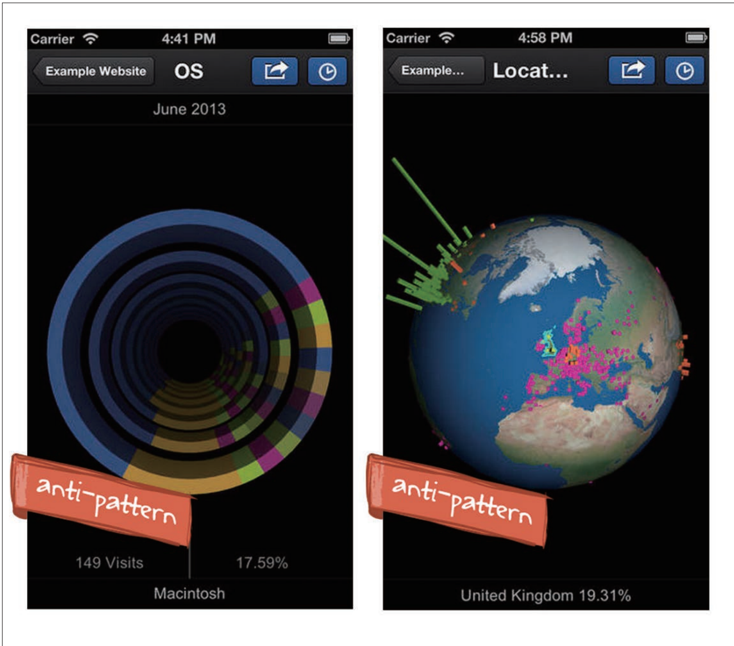
根据维基百科的定义，图表垃圾是“那些不必呈现在图形上的信息，或是那些干扰读者浏览信息的所有图表和图形上的视觉元素”。

注 1：WTFViz 是 Tumblr 上的一个博客，专门收集毫无意义的可视化图表。（译者注）

Analytics 3D 是图表垃圾的典型例子。这些图表上的视觉元素实在太让人分神，甚至连想看清图表上的数据都不可能。但是，嘿，它可是 3D 的啊！

图 11-28

安卓版 Analytics 3D：图表垃圾的典型例子，包含一个饼图 / 虫洞以及一个长满了刺的地球




再来看看 OOKLA Speedtest 应用中反模式的例子，这个闪闪发光的七彩计量器以及流沙效果的蓝色背景都让人不能专心于阅读数据。查看 Cashish 的图表更是一大挑战，它那色彩缤纷的图表如彩虹般躺在一张美钞之上。

甚至像网格线这种看似无害的设计元素也可能是图表垃圾。在两个健康类应用 Glucose Buddy 和 Heart Pal 中，粗壮的水平和垂直网格线实际上让数据更难看清。




图 11-29

安卓版 OOKLA Speedtest 和 iOS 版 Cashish：全是图表垃圾



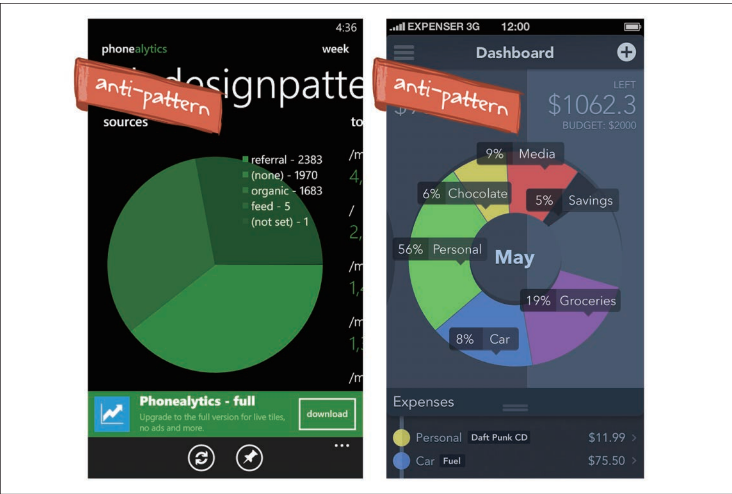

图 11-30

iOS 版 Glucose Buddy 和 Heart Pal：网格线让人无法专注于数据

下面两个例子用来说明页面上堆积的信息太多。在移动应用上表现饼图很困难。由于屏幕空间有限，很难既显示图表又显示图释。

图 11-31

Windows Phone 版 Phonealytics 和 iOS 图表设计概念：饼图对移动 UI 设计是个挑战



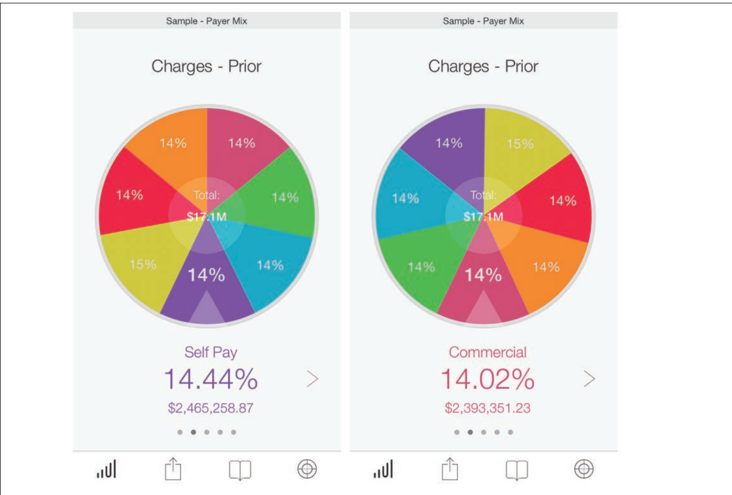

［小贴士］

在图表和图形中，只使用绝对必要的视觉元素传达信息。关于图表设计的最佳实践可参见第 6 章。

然而 Roambi 的设计就兼顾了创新和可用性。你可以旋转图表，且有一个固定的指针显示所指扇区的详情。

图 11-32

iOS 版 Roambi：适用于移动设备的饼图设计（http://youtu.be/WMmMkGF64XY）




## 按钮海

2002 年，在进行 Sabre 航空解决方案（Sabre Airline Solutions）的可用性审查时，我和 Bill Scott 首次命名了按钮海这种反模式。我

们让我们的团队评测了几十个网页端和桌面端应用，以找出最需要重新设计的应用。其中许多应用的界面都使用了 Java Swing 或 Visual Basic 设计，并且在界面的右侧或底部都有一个按钮栏。在底部一行放不下那么多按钮的时候，就会显示两行按钮。所有按钮的大小、颜色、视觉轻重都一样，如果不看完所有按钮上面的字，你根本不知道该点击哪一个。

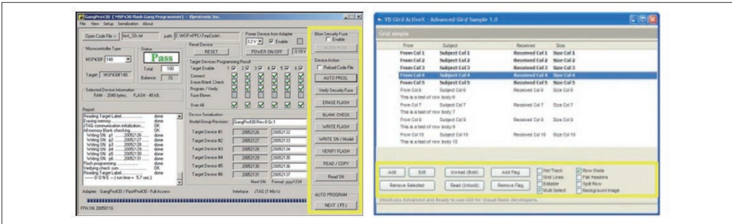


图 11-33
Visual Basic 和 Java 编写的桌面应用：被按钮海淹没

这种反模式中我最喜欢用的一个例子是 Visual KPI。自本书第 1 版出版以来，这个应用又成功添加了更多的控件和按钮！实际上，增加了有一倍那么多。通常我会快速画出一个重新设计的草图，但这个应用需要从信息架构层开始彻底重新设计。

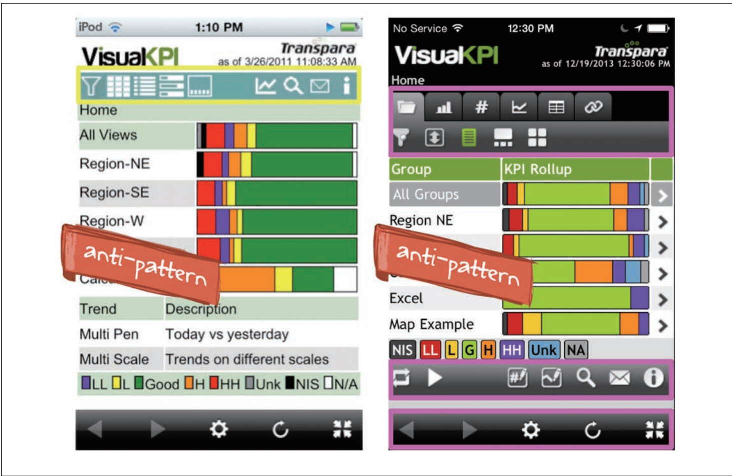
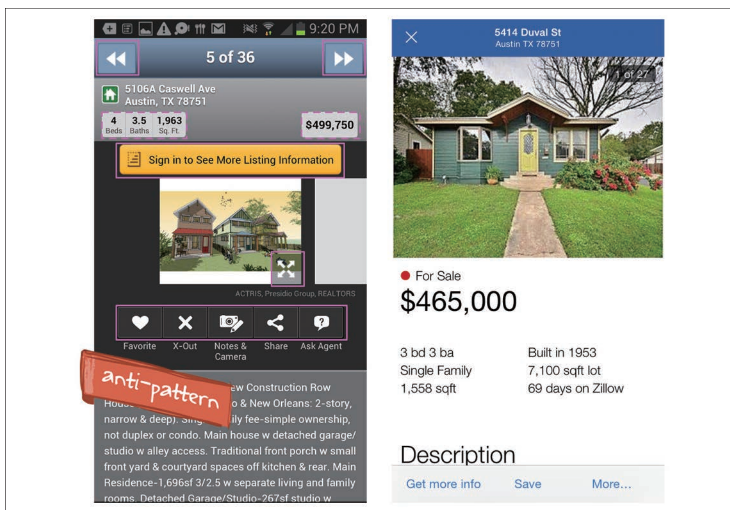



图 11-34
iOS 版 Visual KPI：左图为 2010 年到 2011 年左右的版本，右图为 2013 年左右的版本——现在的按钮更多了！

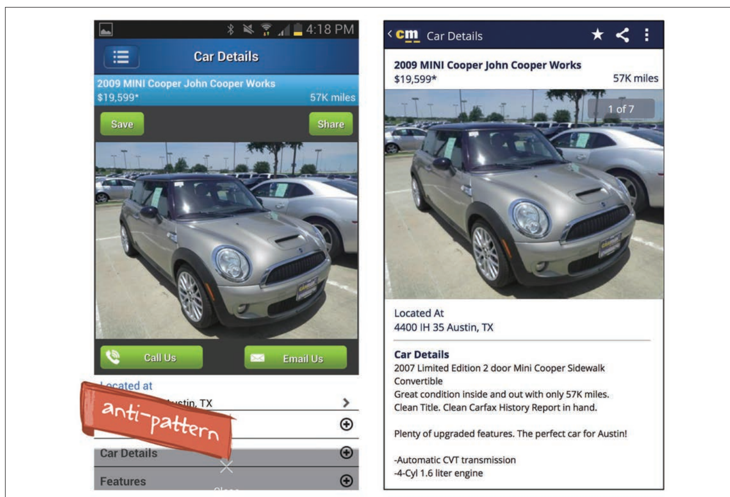
另一个例子是 Redfin。这个应用的页面上有太多按钮（还有一些看上去很像按钮的元素），以致你很难专注于其中的内容（房屋图片及描述）。Zillow 的地产页面就要干净得多，工具栏上显示主要操作，次要操作放进“更多”选项中。

图 11-35
安卓版 Redfin 和 iOS 版 Zillow：按钮海与带有工具栏的干净设计




CarMax 倒是没有按钮海，但是它的“四角按钮”不是好的设计方案。安卓操作栏就可以提供一个简单的修正，像我重新设计的图那样。“保存”和“分享”应该显示出来，而“致电”和“发送电子邮件”选项可以从“更多”菜单进入。

图 11-36
安卓版 CarMax：四角按钮设计效果不佳；右图为我的低保真重新设计版本，使用了安卓操作栏




［小贴士］

使用系统自带的控件来实现页面层级的操作。这样做的好处是，直到出现需要用到按钮的情景之前，它们都不会碍眼。关于按钮的最佳实践参看第 5 章。

Plume 在一条推文被点击时使用了情景工具模式来显示选项。但是，用点击的方式显示工具而不是显示完整的会话系列，会产生

一个交互设计问题。用户怎样才能查看完整的会话呢？要是我的话，就不会按照 Plume 设计的那样又增加一行“查看完整会话”的按钮，我会重新思考整个交互设计。

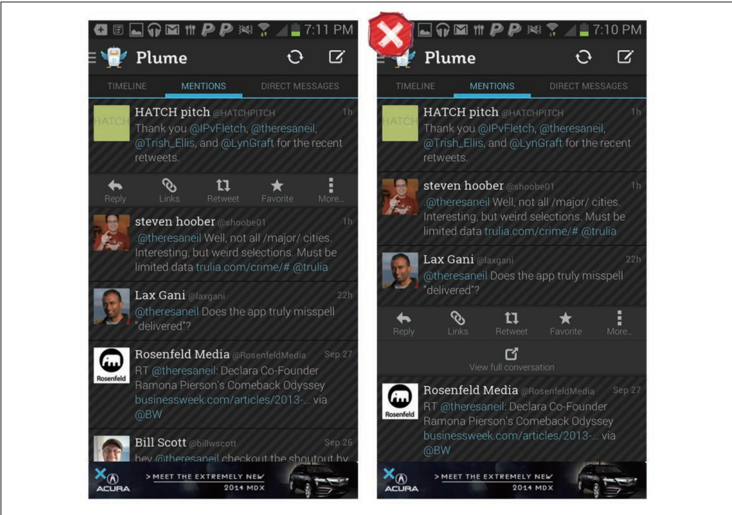



图 11-37

安卓版 Plume：一行按钮还好，但两行按钮就太多了

Tweetbot 在其 iOS 版本的设计中解决了这个问题。它也采用了点击显示情景工具的模式，但是可以向左滑动打开消息详情。




图 11-38

Tweetbot：点击消息显示上下文工具；向左滑动查看消息详情

## 格格不入

格格不入的反模式通常是在执行层面做出无知决定的结果，或是匆忙投入市场的结果，或是两者兼而有之。这种模式是指，采用了一个为单一平台所做的设计，又将其套用到另一个平台当中。

许多公司不做任何修改就把 iOS 平台设计直接搬到安卓平台上去，Behr Paint 正是其中之一。不仅界面看上去搞笑（图形的分辨率错了），而且还打破了安卓设计规范（在 http://developer.android.com/design/patterns/pure-android.html 上查看安卓特定的设计规范列表）。

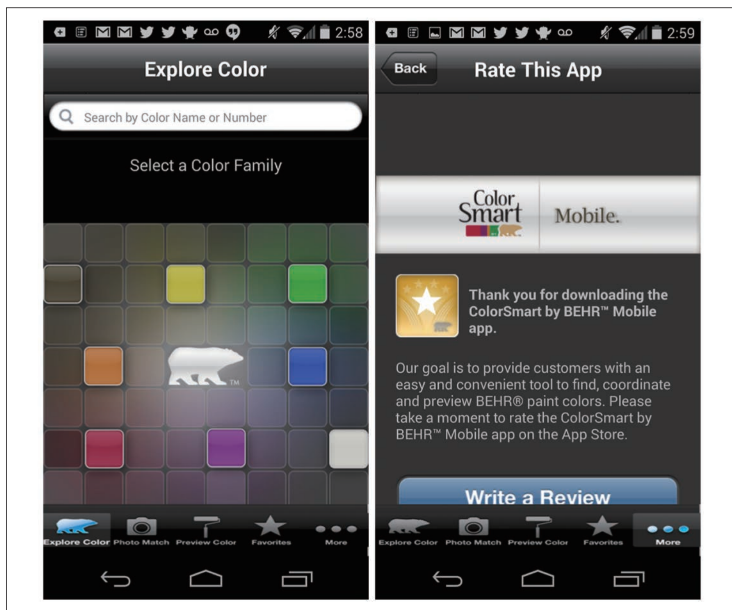



图 11-39

安卓版 Behr Paint：把 iOS 版的设计原封不动地搬到了安卓平台

Gap 应用就是给其移动网站加了个壳，然后就管它叫作“应用”了（http://docs.webplatform.org/wiki/concepts/internet_and_web/html5_hybrid_applications）。但用户可不是傻子，看看用户的评价吧（https://play.google.com/store/apps/details?id=com.skava.hybridapp.gap）。

我认为 SmartyPig 也做出了同样的事情。你能看到这个应用的首页就是一个登录页面，明显没有针对我使用的 Nexus 5 做过适配。我猜这个应用的使用率相当低。

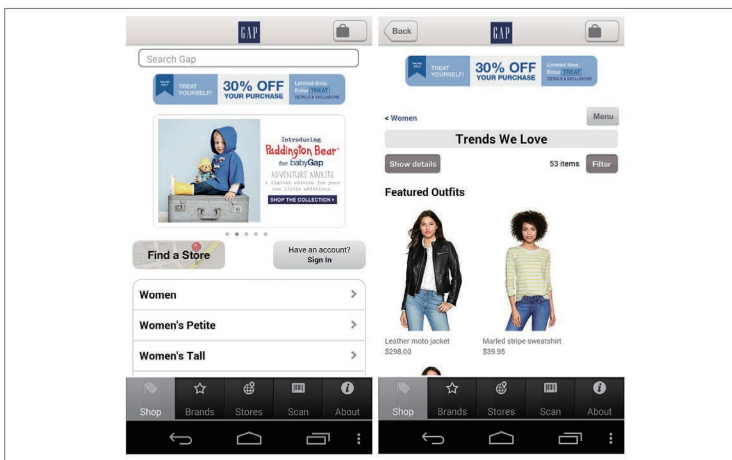



图 11-40
安卓版和 iOS 版 Gap：套了个壳的移动网站

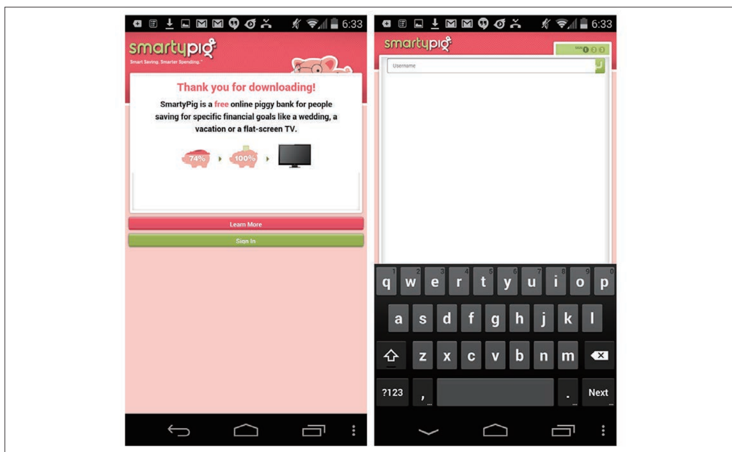



图 11-41
安卓版 SmartyPig

做到以下几点可避免格格不入的反模式：

- 通读并理解目标操作系统的设计指南。
- 每天使用你为之设计 / 开发的移动设备，直到你对那个平台的细节全部了如指掌为止。
- 为每个平台分别设计不同的 UI，遵守它们的设计规范。

还有，小心那些向你承诺“一次编程，全平台发布”的服务。如果你要使用其中一个这样的服务，就选择那个能让你为每个系统创建自定义用户界面的平台，比如 Xamarin（https://xamarin.com/）

[小贴士]

每个平台之间都有明显的区别：一个绝妙的 iOS 设计可能会是安卓平台上的差劲设计，反之亦然。通过设计和测试，为每个平台打造超凡的体验。

或 PhoneGap（http://phonegap.com/）。

## 番外篇：避免移动应用中的注册/登录反模式

作者：Greg Nudelman，Design Caffeine 公司创始人

反模式这一术语出自 Tamara Adlin

摘自《Android 应用 UI 设计模式》（*Android Design Patterns: Interaction Design Solutions for Developers*，2013 年 Wiley 公司出版，由 Boxesandarrows 再版）[1]



在客户下载了你的应用之后，任何让他们放慢速度或妨碍到他们的设计都是不好的设计。这其中就包括，在潜在客户还没能认清这个应用是否真值得使用前，就出现的登录 / 注册表单。

长注册页面与重要的移动用例背道而驰，这个用例是：随时随地快速简单的信息访问。随着公司开始想明白以下这个简单的用户体验等式，这一反模式变得越来越少。这个等式即

用户使用应用前出现长注册表单 = 用户删除应用

然而，还是有很多应用在被使用前就强迫客户注册、登录，或执行一些其他没用的操作。

比如 SitOrSquat 应用，它是一个非常出色的小社交工程软件，可以让人们在想上厕所的时候在街上找到公共厕所。

注 1：中文版已由人民邮电出版社出版。（编者注）

但是收购了这个应用的宝洁公司（Procter & Gamble）似乎完全没有感觉到这种紧急情况的存在。宝洁的高管们不满足于简单的“Squeezing the Charmin”[2]，出于一些莫名的原因，决定以多种方式强迫人们注册应用。

除了注册模块之外的整个应用基本由 4 个页面组成（如果把添加厕所的功能也算上的话！）。然而，如果你把全部反模式的注册页面也包括在内，“预备”阶段的垃圾就用了 7 个页面，在这之后你才能最终看到你想要的内容。如果计算输入生日所必须作出的点击次数，这一数字竟然达到 50 次之多！

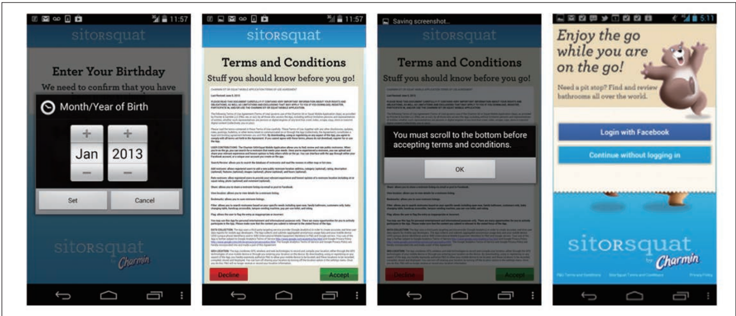



图 11-42
安卓版 Charmin SitrOrSquat 应用（2013 年）

在决定是否强迫客户执行一项操作时，应该考虑：如果这是个网页应用的话，你会强迫客户这么做吗？在互联网连接可用时，你可以保存客户的任何行为，并利用一个简单的会话令牌（session token）和客人账号反向连接到他的设备。即使他不在线，如今的智能手机也拥有足够大的机载存储，在移动网络可用后可以用来与你的服务器进行同步。

这意味着根本没有理由强迫任何人注册任何东西，除非他想要从自己的手机里与其他设备分享数据。作为一条普遍适用的原则，允许客户在其手机本地保存一小部分信息而不用登录，要比在下载后第一次使用时就强迫用户注册好得多。等到客户想要做某些事的时候，比如与其他设备分享信息，或访问已保存在他的账号内的信息，再要求他完成注册就完全合乎情理。但即便如此，也应该只要求用户填写下一步处理确实需要的信息，不应索要与之无关的信息。

注 2：Charmin 为宝洁收购的卫生纸品牌，“Squeezing the Charmin”是其早期一则广告中的广告词。（译者注）

# 附录

## 其他资源

### 网站

http://www.mobiledesignpatterngallery.com

http://www.theresaneil.com

### Flickr

http://www.flickr.com/photos/mobiledesignpatterngallery/sets/

### Twitter

https://twitter.com/theresaneil

TURING 图灵教育

# 更 多 推 荐




书号：978–7–115–34090–0
Patrick McNeil ( 作者 )
定价：49.00 元




书号：978–7–115–34338–3
Robbin Williams ( 作者 )
定价：69.00 元



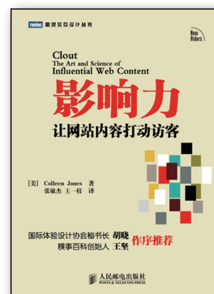

书号：978–7–115–35385–6
长谷川 敦士 ( 作者 )
定价：79.00 元




书号：978–7–115–36343–5
古贺直树 ( 作者 )
定价：39.00 元



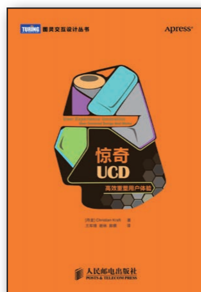

系列书名　图灵交互设计丛书
David Kadavy ( 作者 )
定价：79.00 元



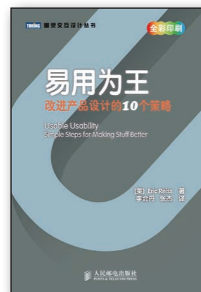

书号：978–7–115–35242–2
Colleen Jones ( 作者 )
定价：49.00 元




书号：978–7–115–32918–9
赵大羽，关东升 ( 作者 )
定价：69.00 元




书号：978–7–115–32339–2
Christian Kraft ( 作者 )
定价：39.00 元




书号：978–7–115–32454–2
Eric Reiss ( 作者 )
定价：59.00 元

**关注图灵教育 关注图灵社区**

# iTuring.cn

在线出版　电子书《码农》杂志　图灵访谈 ……



QQ联系我们

读者QQ群：218139230



微博联系我们

官方账号：@图灵教育 @图灵社区 @图灵新知
市场合作：@图灵袁野
写作本版书：@图灵小花
翻译英文书：@李松峰 @朱巍ituring @楼伟珊
翻译日文书或文章：@图灵乐馨
翻译韩文书：@图灵陈曦
电子书合作：@hi_jeanne
图灵访谈/《码农》杂志：@李盼ituring
加入我们：@王子是好人



微信联系我们



图灵教育
turingbooks



图灵访谈
ituring_interview

O'REILLY®

# 移动应用UI设计模式（第2版）

在你急需设计出一个精美易用的移动应用之时，既来不及也没有必要重新发明轮子。本书通过1000张屏幕截图，为你讲解时下安卓、iOS以及Windows Phone平台上90多种移动应用UI设计模式。

自本书第1版出版以来，移动领域日新月异。移动操作系统之间的区别越来越大，不断驱动着各自平台的设计规范和模式的发展，同时越来越多的设计师也开始秉持以移动为中心的设计理念。在这一版中，用户体验专家Theresa Neil将带领产品经理、设计师和开发者学习11个类别的移动应用UI设计模式。

**本书主要介绍以下设计模式**

- 导航模式：了解主导航和次级导航的设计模式
- 表单模式：改掉行业内表单设计的不良习惯
- 表格模式：只显示最重要的信息
- 搜索、排序和筛选模式：让这些功能易于使用
- 工具模式：创造直接交互的感觉
- 图表模式：了解基本图表设计的最佳实践
- 引导和视觉吸引模式：吸引用户使用应用并发现应用的功能
- 社交模式：帮助用户彼此连接，成为群体中的一份子
- 反馈与功能可见性模式：为用户提供及时的反馈
- 帮助模式：在应用中设计小型的帮助页面
- 反模式：设计移动应用时要规避的设计模式

“移动设计发展飞速，所以看到Theresa Neil的重要著作有新的版本问世，我感到无比激动。第1版中的一系列设计模式已被证明具有极强的适应性，而升级和改进使这本书既实用又与时俱进。我迫不及待要与我的团队分享这些新增的内容。”

——Christian Crumlish
CloudOn公司产品总监

**Theresa Neil** 美国得克萨斯州奥斯汀市的一名用户体验咨询师，国际知名设计专家，致力于打造外观优美、功能完善的产品。她的著作和演讲帮助众多IT从业者提高了自身的设计技能，创造出了更好的用户体验。2013年，她被美国知名的科技博客Business Insider评为“科技领域最优秀的75位设计师”之一，排名第28位。她的网站是：www.theresaneil.com。

封面设计：Randy Comer 张健

图灵社区：iTuring.cn
热线：(010)51095186转600

分类建议 计算机/移动开发

人民邮电出版社网址：www.ptpress.com.cn

O'Reilly Media, Inc.授权人民邮电出版社出版

此简体中文版仅限于中国大陆（不包含中国香港、澳门特别行政区和中国台湾地区）销售发行
This Authorized Edition for sale only in the territory of People's Republic of China (excluding Hong Kong, Macao and Taiwan)




ISBN 978-7-115-37790-6
定价：79.00元

# 看完了

如果您对本书内容有疑问，可发邮件至contact@turingbook.com，会有编辑或作译者协助答疑。也可访问图灵社区，参与本书讨论。

如果是有关电子书的建议或问题，请联系专用客服邮箱：ebook@turingbook.com。

在这里可以找到我们：

微博 @图灵教育 : 好书、活动每日播报
微博 @图灵社区 : 电子书和好文章的消息
微博 @图灵新知 : 图灵教育的科普小组
微信 图灵访谈 : ituring_interview，讲述码农精彩人生
微信 图灵教育 : turingbooks